



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

**NÁVRH A IMPLEMENTACE NOVÉHO MODULU
PODNIKOVÉHO WEBOVÉHO NÁSTROJE PRO SPRÁVU
PROJEKTŮ**

DESIGN AND IMPLEMENTATION OF A NEW MODULE FOR COMPANY WEBTOOL FOR PROJECTS
ADMINISTRATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Petr Hladík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2017

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Petr Hladík**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh a implementace nového modulu podnikového webového nástroje pro správu projektů

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je navrhnout a vytvořit nový modul pro podnikový webový nástroj, který využívá vybraná společnost jako svůj informační systém se zaměřením na správu projektů.
Výstupem bude přehledný nástroj pro správu domén klientů a jeho automatické provázání na projekty se zaměřením na podporu automatizace správy a funkčního provázání na informační systém společnosti.

Základní literární prameny:

BRUCKNER, T., J. VOŘÍŠEK, A. BUCHALCEVOVÁ a kol. Tvorba informačních systémů: Principy, metodiky, architektury. Praha: Grada Publishing, 2012. 360 s. ISBN 978-80-247-4153-6.

CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. Mistrovství – databáze: Profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KOFLER, M a B. ÖGGL. PHP 5 a MySQL 5: Průvodce webového programátora. 1. vyd. Brno: Computer Press, 2007. 608 s. ISBN 978-80-251-1813-9.

KOSEK, J. PHP a XML. 1. vyd. Praha: Grada, 2009. 368 s. ISBN 978-80-247-1116-4.

VRÁNA, J. 1001 tipů a triků pro PHP. 1. vyd. Brno: Computer Press, 2010. 456 s. ISBN 978-80-2-1-2940-1.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně dne 28.2.2017

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Cílem této práce je návrh modulu pro informační systém společnosti XART, s.r.o., zaměřující se na vytváření webových řešení na míru a internetový marketing. Primární funkčnosti modulu jsou automatické získávání, ukládání a prezentace informací o klientských webech. Díky rychlému poskytnutí těchto informací prostřednictvím modulu je pak zaměstnanec schopen zvýšit svoji produktivitu práce a to kvůli snížení potřebného času na dohledávání informací o konkrétním klientském webu.

Abstract

The aim of the bachelor's thesis is design of a module for information system for a XART, s.r.o. company. The company is focused on development of special web solutions and internet marketing. The module's primary functionalities are automatic downloading, saving and presenting informations about client's websites. The module is able to increase employee productivity by reducing time needed to search for information about client's websites.

Klíčové slova

Informační systém, PHP, OOP, API, MySQL, databáze, Virtualmin

Key words

Information system, PHP, OOP, API, MySQL, database, Virtualmin

Bibliografická citace

HLADÍK, P. *Návrh a implementace nového modulu podnikového webového nástroje pro správu projektů*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 96 s.
Vedoucí diplomové práce Ing. Jan Luhan, Ph.D., MSc.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 31. května 2017

.....

podpis studenta

Poděkování

Tímto bych chtěl poděkovat všem, kteří mi pomáhali při zpracovávání této práce, hlavně těm, kteří mi poskytovali cenné rady a informace, zejména vedoucímu práce Ing. Janu Luhanovi, Ph.D., MSc.

OBSAH

ÚVOD.....	10
1 CÍL A METODIKA PRÁCE	11
2 TEORETICKÁ VÝCHODISKÁ PRÁCE	12
2.1 Operační systém	12
2.2 Příkazový řádek	12
2.3 Internet	13
2.3.1 World Wide Web	13
2.3.2 IP adresa.....	13
2.3.3 Domain name systém – DNS	14
2.3.4 Hypertext transfer protocol – HTTP	15
2.3.5 Webový prohlížeč	15
2.3.6 Webová stránka.....	15
2.3.7 Hypertext	16
2.4 Hypertext markup language – HTML	16
2.5 Extensible Markup Language – XML.....	16
2.6 Hypertext preprocessor – PHP	17
2.6.1 Interpretace jazyka	17
2.6.2 Možnosti jazyka	18
2.7 JavaScript	18
2.8 JavaScript Object Notation – JSON	19
2.9 Document Object Model – DOM.....	19
2.10 Asynchronous JavaScript and XML – AJAX.....	19
2.11 Objektové programování	20
2.11.1 Objekt	20
2.11.2 Atribut	20
2.11.3 Metody.....	21
2.11.4 Třída	21
2.12 Framework	21
2.13 MVC architektura.....	22
2.13.1 Model.....	23
2.13.2 View	23
2.13.3 Controller.....	24
2.14 Databáze	24
2.14.1 Databázové modely	24
2.14.2 Relační databázový model MySQL databáze	25
2.14.3 Typy databázových tabulek MySQL databáze	26
2.15 Webová služba	27
2.16 SOAP.....	27
2.17 Webmin	27
2.18 Virtualmin	27
2.19 Informační systém.....	28
2.19.1 Důvody k provozu informačního systému	28
2.19.2 Obsah informací v informačním systému organizace	29
2.19.3 Základní funkce informačního systému.....	30
2.19.4 Základní členění podnikových aplikací	31
2.20 Active 24, s.r.o.	32
3 ANALÝZA PROBLÉMU A SOUČASNÁ SITUACE	33
3.1 ZÁKLADNÍ INFORMACE O SPOLEČNOSTI	33
3.1.1 Přehled základních informací.....	33
3.1.2 Organizační podmínky podniku	34
3.2 Nabízené produkty a služby společnosti	34
3.2.1 Internetový marketing	34

3.2.2	Tvorba www prezentací	35
3.2.3	Programování zakázkových aplikací	36
3.2.4	Virtualizace serverů a desktopů	36
3.3	Technologické vybavení	37
3.3.1	Konfigurace produkčního serveru	37
3.3.2	Produkční dělení disků na kontejnery	38
3.4	Podnikový informační systém	38
3.4.1	Konfigurace serveru pro provoz informačního systému	39
3.4.2	Doporučené požadavky na klienta pro využívání Active Collab verze 4.1.7	40
3.4.3	Využívané moduly systému	41
3.4.4	Práva v prostředí systému	42
3.4.5	Povolené role v systému	43
3.4.6	Pátevní adresářová struktura souborů systému	44
3.4.7	Pátevní aplikační struktura systému	45
3.4.8	Překlady rozšíření	46
3.5	XART Wiki	47
3.6	Specifikace problematiky	48
4	Návrh řešení	51
4.1	Základní požadavky na modul	51
4.2	Postup při aktualizaci dat	52
4.2.1	Aktualizace seznamu domén	54
4.2.2	Aktualizace konfigurace domén	55
4.2.3	Aktualizace rozšíření webů	57
4.2.4	Aktualizace DNS záznamů domén	60
4.2.5	Aktualizace konfigurací plánů domén	63
4.3	Datový model	64
4.3.1	Relace „Domény“ (#__domains)	65
4.3.2	Relace „Konfigurace domény“ (#__domains_server_configuration)	66
4.3.3	Relace „Databáze domény“ (#__domains_databases)	67
4.3.4	Relace „DNS“ (#__domains_dns)	68
4.3.5	Relace „SOA DNS“ (#__domains_dns_soa)	69
4.3.6	Relace „Konfigurace plánů“ (#__domains_plans)	71
4.3.1	Relace „Rozšíření domény“ (#__domains_extensions)	71
4.3.2	Relace „Historie změn“ (#__domains_histories)	72
4.3.3	Relace „Vazby na projekty“ (#__domains_related_projects)	73
4.4	Implementace modulu	75
4.4.1	Základní vlastnosti	75
4.4.2	Oprávnění	76
4.4.3	Harmonogram prací pro vytvoření modulu	77
4.5	Vizualizace uživatelského prostředí modulu	79
4.6	Ekonomické zhodnocení	84
4.6.1	Náklady	84
4.6.2	Investice do rozšíření informační systému	86
4.7	Přínosy pro podnik	87
	ZÁVĚR	88
	SEZNAM POUŽITÝCH ZDROJŮ	89
	SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	93
	SEZNAM OBRÁZKŮ	94
	SEZNAM TABULEK	95
	SEZNAM PŘÍLOH	96

ÚVOD

V současnosti si život a práci v mnoha oborech bez informačních systémů dokážeme jen těžko představit, protože usnadňují a zefektivňují každodenní práci.

Informační systém může například řídit vnitropodnikové každodenní rutinní úkony, objednávat import a export zboží, zadávat úkoly zaměstnancům, efektivně plánovat pracovní směnu atd. Dokáží v mnoha případech efektivně šetřit zdroje (čas a peníze) a zvýšit produktivitu.

Tyto informační systémy je potřeba udržovat, aktualizovat a rozšiřovat dle potřeb, které se mohou během existence podniku měnit v závislosti na oboru podnikání. Informační systém, který splňuje všechny požadavky podniku, může přispět ke zvýšení konkurenceschopnosti.

Podniky, které si tuto skutečnost uvědomují, jsou poté ochotny investovat finanční prostředky pro udržování a rozvoj vlastního informačního systému.

V mé práci se budu věnovat návrhu nového modulu informačního systému, který bude navržený na míru podle konkrétních potřeb podniku vzhledem k oboru působení tak, aby pokryl funkcionalitu, která v informačním systému podniku chybí.

V teoretických východiskách práce popíšu základní informace, která jsou potřeba pro pochopení dalších částí práce. Následně se zaměřím na analýzu současného stavu, kde uvedu informace o podniku a jeho informačním systému. Poté se budu věnovat samotnému návrhu modulu a jeho implementaci do systému. V této části se budu nejvíce zabývat procesem aktualizace dat, který je pro modul klíčovým. Součástí návrhu řešení bude i zachycení ekonomických hledisek práce.

1 CÍL A METODIKA PRÁCE

Cílem práce je navrhnout a vytvořit nový modul (přehledný nástroj) pro podnikový webový nástroj pro správu projektů, který podnik používá jako podnikový informační systém. Tento modul má za úkol propojit webové domény klientů, které podnik spravuje, s projekty v projektovém manažeru a to tak, že sám bude schopen detekovat a zobrazit skutečný stav hostingu (typ a verze CMS, jeho komponenty a moduly, rozšíření systému, řešení na míru, propojení s API Active 24, s.r.o. a interním SVN).

Modul bude napsán v programovacím jazyce PHP (stejně jako celý informační systém, do kterého je nutné modul implementovat). Bude brán ohled na výkon celého řešení, protože podnik spravuje desítky webů a automatická detekce aktuálního stavu webů musí být co nejlépe procesně optimalizována. Musí být tedy navržen takový algoritmus aktualizace, který dokáže efektivně (rychle) získávat data z externích systémů, tato data zpracovat a uložit.

Motivací pro vytvoření modulu je skutečnost, že webů je velké množství a není zatím zavedena žádná metodika, jak efektivně centralizovaně uchovávat informace o aktuálním stavu webu.

2 TEORETICKÁ VÝCHODISKÁ PRÁCE

V této části jsou shrnuty základní pojmy a informace, které jsou klíčové pro pochopení dalších částí práce a vytvoření modulu pro konkrétní informační systém podniku.

Jsou zde vysvětleny i témata, na která se v dalších částech práce přímo neodkazuje, ale jsou podstatná pro pochopení ostatních teoretických základů nebo pro vytvoření uceleného pohledu na danou problematiku.

2.1 Operační systém

Operační systém představuje nezbytné softwarové vybavení počítače, které slouží k ovládání hardwaru. Hlavním úkolem operačního systému je přidělování prostředků hardwaru počítače jednotlivým procesům spuštěným v rámci operačního systému. V současnosti existuje několik různých typů systémů pro různá zařízení a licencemi pro používání (1).

2.2 Příkazový řádek

CLI (Command line interface – příkazový řádek) je uživatelské textové rozhraní operačního systému nebo aplikace, kde uživatel zadává příkazy a systém nebo aplikace je interpretují do programového příkazu, zpracují a navrátí výsledek. Mechanismus je velice jednoduchý, ale na rozdíl od grafického rozhraní není pro uživatele příliš přívětivý (2).

2.3 Internet

Internet je celosvětová síť (desítky tisíc jednotlivých sítí – možná i mnohem více) propojující všechny kontinenty, státy a města tak, že jakýkoliv počítač může kdykoliv komunikovat s jakýmkoliv jiným počítačem, ať je kdekoliv na planetě. Internet nikdo neřídí, funguje díky tomu, že všichni lidé mají společný zájem – chtějí být propojeni (3).

2.3.1 World Wide Web

World Wide Web je dnes nejpoužívanější služba internetu. Zpravidla zkracuje na „www“. Díky vzniku této služby se internet masově rozšířil do společnosti. Službu stvořili Tim Berners-Lee a Robert Cailliau. Využili principu hypertextu – souboru textů navzájem propojených odkazy – a přidali k tomu komunikační protokol HTTP (hypertext transfer protocol) (4).

2.3.2 IP adresa

Aby mohla zařízení v internetu vzájemně komunikovat, je potřeba komunikaci adresovat. Pro tento účel vznikly IP adresy, které umožňují síťová zařízení v síti používající internetový protokol jednoznačně identifikovat.

Hodnoty IP adres se zapisují buď v desítkové, nebo hexadecimální soustavě s použitím oddělovačů (o způsobu zápisu a oddělení hodnot rozhoduje verze IP adresy). Existují momentálně dvě používané verze IP adres:

- IPv4 – 32 bitové adresy,
- IPv6 – 128 bitové adresy (5).

2.3.3 Domain name system – DNS

IP adresy nejsou vhodné pro zapamatování, proto byly vytvořeny systémy DNS, které umožňují konkrétní IP adrese (tedy síťovému zařízení) přiřadit tzv. „doménové jméno“ (textový řetězec), které si uživatelé internetu snadněji zapamatují (6).

V systému DNS platí určitá hierarchie. Doménová jména jsou rozdělena na několik úrovní, která se oddělují tečkou a celý zápis za sebou tvoří plně kvalifikované doménové jméno. Jednotlivé úrovně jsou číslovány pozpátku:

- první část jména je doména nejvyšší úrovně – TLD (top level domain),
- následuje druhá část jména – SLD (second level domain),
- dále následují nepovinné úrovně (třetí, čtvrtá, atd.), které nejsou speciálně označovány (6).

Chod služeb na doméně ovlivňuje nastavení DNS záznamů (7).

Existuje několik typů DNS záznamů, například:

- „A“ – nastavení IP adresy IPv4, na kterou se má doménové jméno překládat,
- „AAAA“ – alternativa k „A“ záznamu, ale v IPv6,
- „MX“ – nastavení serveru, kam se bude zasílat elektronická pošta zasílána na e-mailové adresy domény,
- „CNAME“ – nasměrování domény na libovolnou jinou doménu,
- „TXT“ – zadávání libovolných textových řetězců (například pro ověření vlastnictví domény atd.),
- „SPF“ – specifický typ TXT záznamu, kterým se ověřuje odesílatel elektronické pošty (konkrétně ověřuje, zda poštovní server odesílatele je oprávněn odesílat poštu pro danou doménu) (8).

2.3.4 Hypertext transfer protocol – HTTP

„Http protokol je způsob, kterým si povídá prohlížeč se serverem, když se stahují stránky“
(9).

Obecně se jedná o jednoduchý aplikační bezstavový protokol modelu požadavek/odpověď, pro přenos objektů libovolného typu (stránky, obrázky atd.) mezi webovým serverem a prohlížečem (10).

2.3.5 Webový prohlížeč

Webový prohlížeč je program v počítači, který dokáže zobrazit webovou stránku. Existuje jich několik, např. Internet Explorer, Mozilla Firefox, Chrome, Safari, Opera a další (11).

2.3.6 Webová stránka

Webová stránka je označení dokumentu s informacemi, který je dostupný pomocí internetu a webového prohlížeče a pro návštěvníka webové stránky je prezentován estetickým a čitelným způsobem (12).

Webové stránky se používají pro různé účely, jak pro individuální použití, tak firemní a mohou být buď statické, nebo dynamicky generované (13).

Statická webová stránka je stránka, kde je veškerý obsah pevně kódován do samotné stránky. Tyto stránky jsou obvykle kódované pouze za použití HTML (HyperText Markup Language) (13).

Dynamické webové stránky jsou generovány skriptem, který sám sestavuje HTML dokument. Tento dokument může vygenerovat například za použití dat z databáze.

Dynamické webové stránky lze vytvářet za použití několika programovacích jazyků, včetně PHP (13).

2.3.7 Hypertext

„Text, ve kterém fungují některé výrazy jako odkazy. Prostřednictvím těchto odkazů se přenesete na jinou část dokumentu nebo na úplně nový dokument“ (14).

2.4 Hypertext markup language – HTML

HTML je značkovací jazyk pro hypertext. Je jedním z jazyků, které umožňují vytvářet dokumenty, které je pak možné publikovat na internetu. Je definován množinou značek a jejich atributů. Mezi značky se uzavírají části textu a tím se určuje význam obsaženého textu. Jednotlivé značky se uzavírají mezi úhlové závorky „<“ a „>“. Součástí elementu mohou být v obsahu další vnořené elementy. Pomocí atributů se elementu doplňují informace, které definují vlastnosti elementu (15).

Značky neboli „tagy“ mohou být párové a nepárové. Párové značky se uzavírají koncovou značkou, která je shodná s počáteční, akorát se před názvem vpisuje znak lomítka „/“ (15).

2.5 Extensible Markup Language – XML

XML (česky „rozšiřitelný značkovací jazyk“) je jednoduchý a velmi flexibilní textový formát, který se používá pro popis dat. Je to flexibilní způsob, jak vytvořit strukturovaná data a snadno je sdílet prostřednictvím sítě (16).

Je velmi podobný formátu HTML, oba jazyky jsou složeny ze značkovacích značek a oba slouží k popisu obsahu. XML na rozdíl od HTML je tzv. „samodefinující“, což znamená, že struktura

dat je uložena s daty, díky tomu není nutné předem vytvářet strukturu souboru pro uložení dat (16).

Formát XML je tedy velmi vhodný pro výměnu informací mezi prostředími, které vyžadují komunikaci konzistentním způsobem (16).

2.6 Hypertext preprocessor – PHP

„PHP je programovací jazyk, který pracuje na straně serveru. S PHP můžete ukládat a měnit data webových stránek. Původní význam zkratky PHP byl Personal Home Page. Vzniklo v roce 1996, od té doby prošlo velkými změnami a nyní tato zkratka znamená PHP: Hypertext Preprocessor” (17).

Oficiální stránky PHP se nacházejí na www.php.net (20).

Od verze 5, která byla vydána 13. července 2004, PHP funguje nad novým skriptovacím engine (virtuálním strojem) Zend Engine II, díky kterému bylo v PHP umožněno na rozdíl od verze 4 efektivnější objektové modelování (18).

2.6.1 Interpretace jazyka

PHP je imperativní (procedurální) interpretovaný multiparadigmatický programovací jazyk (19).

Imperativní programovací jazyk je jazyk, který k řešení úlohy tedy využívá algoritmu (postup, jak se má daná úloha vyřešit) (19).

Interpretovaný programovací jazyk je jazyk, který je do strojového kódu překládán až za běhu programu (tím se liší od kompilovaných, kde se před spuštěním musí zdrojový kód překompilovat pomocí kompilátoru do strojového kódu a až potom spustěn). Zdrojový

kód se za běhu překládá pomocí interpretu a ten příkazy při překladu zároveň provádí (po částech) (20).

Multiparadigmatický programovací jazyk je jazyk, který podporuje více jak jedno programovací paradigma (19).

Programovacím paradigmatem si lze představit základní programovací styl, ve kterém je program naprogramován. Rozdíl je v pojmech a abstrakcích, které tvoří prvky programu (objekty, funkce, proměnné, omezení atd.). Programovací paradigma rozlišujeme na: aspektově orientované, deklarativní (funkcionální, logické), generické, imperativní, paralelní a strukturované (objektově orientované, rekurzivní) (21).

2.6.2 Možnosti jazyka

PHP má spoustu možností pro práci, dokáže pracovat s databázemi, s velkým množstvím internetových protokolů (např. IMAP, POP3, SMTP atd.), dokáže vytvářet (editovat) soubory a obsahuje spoustu funkcí na zpracování řetězců. PHP je zdarma a volně šiřitelné (20).

2.7 JavaScript

JavaScript je multiplatformní a objektově orientovaný, odlehčený skriptovací jazyk, který se spouští na hostovaném zařízení (např. webový prohlížeč). Může se vázat na objekty hostovaného prostředí a může tak poskytovat prostředky pro jejich ovládání (22).

Neobsahuje statické typování proměnných, ani silnou typovou kontrolu jako v některých jiných programovacích jazycích (22).

Umožňuje manipulovat s prvky webové stránky pro její oživení. Je tedy základem dynamického webu (23).

2.8 JavaScript Object Notation – JSON

JSON je odlehčeným, na programovacím jazyce zcela nezávislým, formátem pro výměnu dat. Díky své jednoduchosti je snadný pro čtení a zápis člověkem i strojově. Zakládá se na dvou strukturách:

- kolekce párů názvů a hodnot (ve většině jazycích realizováno jako objekt – vlastnosti odpovídá hodnota),
- seřazený seznam hodnot (ve většině jazycích realizováno jako pole – seznam hodnot) (24).

2.9 Document Object Model – DOM

DOM je platformě a jazykově neutrální rozhraní, které umožňuje programům a skriptům dynamicky přistupovat a aktualizovat obsah, strukturu a styl dokumentů. Tyto dokumenty mohou být dále zpracovávány a výsledky tohoto zpracování mohou být začleněny zpět do dokumentu (25).

2.10 Asynchronous JavaScript and XML – AJAX

AJAX (česky „asynchronní JavaScript a XML“) je populární způsob pro tvorbu vysoce interaktivních aplikací. Základní princip je aktualizovat pouze části stránky bez nutnosti načítání a vykreslování celé stránky (26).

„AJAX však dovoluje poslat pomocí JavaScriptu separátní HTTP požadavek a data, která nám server vrátí, pak můžeme v JavaScriptu libovolně dále zpracovávat – např. je pomocí DOM zaintegrovat do právě zobrazované stránky. Lze tak interaktivně aktualizovat pouze část stránky” (26).

2.11 Objektové programování

„Objektově orientované programování (dále jen OOP) nevzniklo náhodou, ale je důsledkem vývoje, který k němu směřoval. Jedná se o moderní metodiku vývoje softwaru, kterou podporuje většina programovacích jazyků. Častou chybou je, že se lidé domnívají, že OOP se využívá pouze při psaní určitého druhu programů a jinak je na škodu. Opak je pravdou – OOP je filosofie, je to nový pohled na funkci programu a komunikaci mezi jeho jednotlivými částmi. Mělo by se používat vždy, ať už děláme malou utilitku nebo složitý databázový systém. OOP není jen technika nebo nějaká doporučená struktura programu, je to hlavně nový způsob myšlení, nový náhled na problémy a nová éra ve vývoji softwaru” (27).

Objektem je myšlen soubor dat a funkcí, čímž je patrný velký rozdíl objektového a strukturovaného programování, kdy strukturované od sebe přísně odděluje data a funkce (28).

2.11.1 Objekt

Je základní jednotkou, která dokáže udržovat svůj stav a integrovat s okolím zasíláním a přijímáním zpráv (29).

2.11.2 Atribut

Atributy jsou vlastnosti neboli data (proměnné), která objekt v sobě uchovává. Někdy se o nich hovoří jako o vnitřních stavech objektu (30).

2.11.3 Metody

Metody jsou schopnosti, které je objekt schopen vykonávat. Mohou mít parametry a mohou vracet hodnotu stejně jako funkce. Metoda se od funkce liší tím, že metoda patří objektu, zatímco funkce nepatří žádnému objektu (to je také problém neobjektového programování) (30).

2.11.4 Třída

Aby bylo možné vytvořit objekt, je nutné nejdříve vytvořit třídu. Třída je vzor, podle kterého se objekty vytvářejí, definuje vlastnosti a schopnosti. Objekt, který se vytvoří podle třídy, se nazývá instance. Instance mají stejné rozhraní jako třída, podle kterého se vytvoří, ale navzájem se liší svými atributy (30).

2.12 Framework

Jedním z hlavních společných cílů při programování by měl být šetřit si práci. Toho dosahujeme použitím správných návrhů funkcí, metod a tříd, které lze uplatnit na více místech. Tohoto cíle lze také dosáhnout použitím existujících osvědčených a optimalizovaných knihoven a nástrojů (31).

Framework je ucelený soubor tematicky zaměřených knihoven, které ulehčují práci při programování aplikací. To zajišťuje psaní méně kódu, kód je přehlednější a rychlejší vývoj aplikace. Většina frameworků pracuje s architekturou MVC, která rozděluje aplikaci na nezávislé vrstvy (32).

„Z praktických zkušeností vám mohu s klidem říci, že pokud programujete v čistém PHP, tak je asi 50 % vašeho kódu nepotřebný balast. S použitím kvalitních knihoven napíšete aplikaci za polovinu času a s polovičním počtem řádků. Aplikace bude snadno udržovatelná a její tvorba vás bude bavit. Nebudete stále vynalézat kolo a řešit ty samé

rutinní záležitosti (jako např. jak ověřit, zda je uživatel administrátor), budete se naopak věnovat vymýšlení zajímavých funkcí té dané aplikace a o tom programování přeci je“ (33).

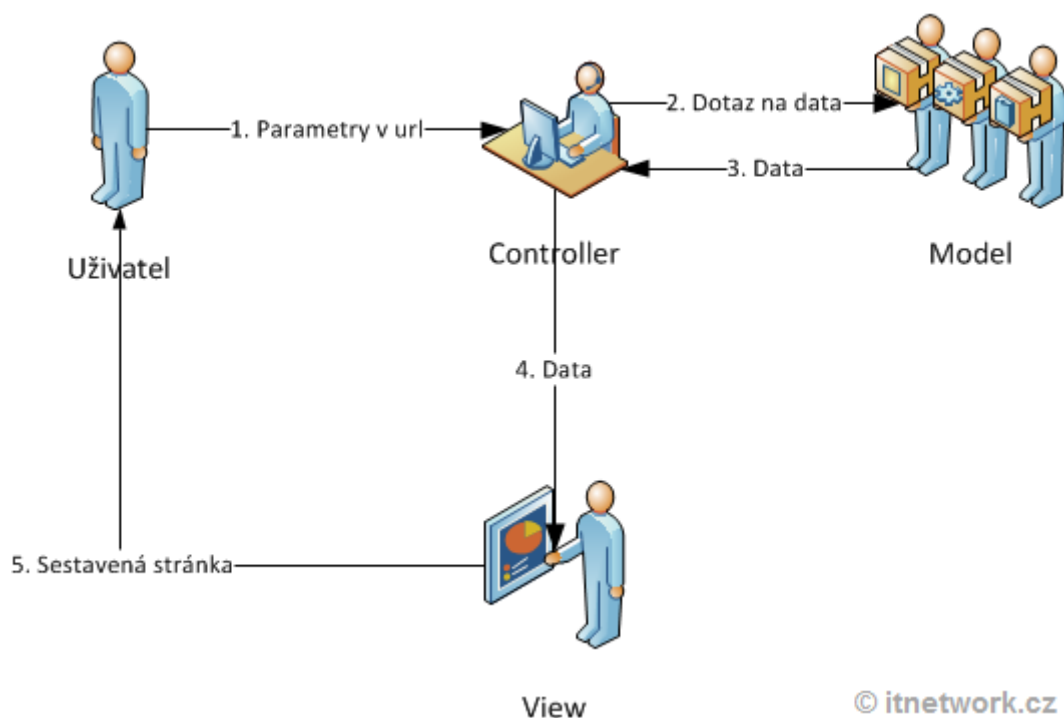
2.13 MVC architektura

MVC architektura neboli architektura Model-View-Controller, je architektura budování aplikace, která oddělí aplikaci na 3 logické části (15).

„Základní myšlenkou MVC architektury je oddělení logiky od výstupu. Řeší tedy problém tzv. "špagetového kódu", kdy máme v jednom souboru (třídě) logické operace a zároveň renderování výstupu. Soubor tedy obsahuje databázové dotazy, logiku (např. PHP operace) a různě poházené HTML tagy. Vše je zamotané do sebe jako špagety. Kód se samozřejmě špatně udržuje, natož rozšiřuje“ (34).

Aplikace se rozdělená na komponenty 3 typů (3 logické části) – Model, View (Pohled) a Controller (nepřekládá se, ale dalo by se přeložit jako „prostředník“) (34).

Na následujícím diagramu je znázorněn životní cyklus dynamicky vygenerované webové stránky s použitím architektury MVC.



Obr. 1: Životní cyklus dynamicky vygenerované webové stránky s použitím architektury MVC, zdroj: (34)

2.13.1 Model

Model obsahuje logiku aplikace. Mohou to být například výpočty, dotazy na databázi, validace apod. Model neví nic o výstupu. Jeho funkce je přijetí parametrů, zpracování a vydání dat ven. Model neví, jak se k němu data dostala a jak budou dále zpracovaná data interpretována (34).

2.13.2 View

Pohled (View) se stará o interpretaci výstupu aplikace. Šablony lze samozřejmě vkládat do sebe, abychom eliminovaly opakování stejného kusu kódu v několika šablonách (34).

2.13.3 Controller

Controller (nepřekládá se) zastává funkci prostředníka, se kterým komunikuje uživatel, model i pohled. Drží systém pohromadě a propojuje jednotlivé komponenty (34).

2.14 Databáze

„Databáze neboli datová základna (Data Base) je místo, kam se ukládají určitým způsobem organizované a strukturované údaje. Přístup k údajům uložených v databázi obstarává databázový systém, kterému se říká SŘBD – Systém Řízení Báze Dat (anglický termín DBMS – Database Management System). Běžně se označením databáze myslí jak uložená data, tak i software“ (35).

Slovem databáze běžně označujeme každý uspořádaný soubor dat (28).

Databáze slouží primárně jako nástroj pro snadnější a efektivnější extrakci informací. Díky uspořádání se z dat stává přehledný systém informací, ze kterého se dají snadno čerpat informace a znalosti (36).

2.14.1 Databázové modely

Existuje několik základních modelů databází:

- lineární model,
- hierarchický model,
- síťový model,
- relační model,
- objektový model (36).

Lineární model – množina souborů záznamů, seřazené dle předmětu zájmu. Mezi jednotlivými záznamy neexistují vazby (36).

Hierarchický model – množiny souborů záznamů, které obsahují vazby nadřazenosti a podřazenosti (tedy rodič a potomek). Pomocí ukazatelů jsou pak znázorněny samotné vazby. Bohužel, tento model má své velké omezení – značně omezuje možnost pohledu na data, protože jediný možný pohled vazbami je možný ve směru pouze rodič – potomek. Není tedy možné podívat se na rodiče skrz potomka (36).

Síťový model – obdoba hierarchického modelu, ale jsou odstraněny nedostatky omezující vazby pouze z rodiče na potomka, je tedy možné vytvářet vazby libovolně mezi segmenty (36).

Relační model – současně se považuje za nejpoužívanější databázový model. Princip modelu spočívá ve využití spojení několika lineárních modelů, které jsou navzájem svázány relačními klíči. Tyto spojení jsou nekonstantní, vznikají pouze na dotaz uživatele. Každý segment obsahuje vlastní jedinečný identifikační klíč, přes který se vytváří spojení s dalším segmentem (36).

Objektový model – model obsahuje objekty a nad těmito objekty je možné definovat metody (např. nový záznam). Samotné metody lze poté považovat jako instance objektu. Díky použití jedinečných identifikátorů objektů pak lze mezi objekty vést vazby, obdobně jako v síťovém nebo relačním modelu (36).

2.14.2 Relační databázový model MySQL databáze

MySQL databáze se skládá z několika tabulek. Jednotlivé tabulky se skládají z řádků, které se označují jako záznam. Struktura záznamu je složena z polí (sloupců) a je určena definicí tabulky. Uvnitř databázové tabulky se mohou vyskytovat odkazy na jiné tabulky, tyto odkazy se označují jako relace (28).

„Popis databáze obsahující několik tabulek se všemi jejími poli, relacemi a indexy označujeme jako schéma databáze. Toto schéma tedy určuje strukturu dat a zároveň určuje i formát, v němž se vlastní data ukládají“ (28).

Díky standardu SQL (Structured Query Language) lze formulovat dotazy na databázi a díky tomu lze s databází manipulovat. U rozsáhlých databázových tabulek závisí rychlost zpracování dotazu na tom, zda se používá vhodný index (klíč). Indexem se rozumí další tabulka, která obsahuje informace o pořadí záznamů (37).

2.14.3 Typy databázových tabulek MySQL databáze

Existuje několik typů databázových tabulek MySQL databáze, které se navzájem liší svými vlastnostmi:

- MyISAM,
- InnoDB,
- HEAP (28).

Tabulky typu MyISAM – zcela standardní, stabilní, vespělý a jednoduše spravovatelný typ tabulek. Pokud nejsou na tabulku žádné zvláštní požadavky, doporučuje se používat právě tento typ tabulek (28).

Tabulky typu InnoDB – podporuje všechny vlastnosti typu tabulek MyISAM, ale na rozdíl od MyISAM podporuje používání pravidel integrity (tzv. pravidlo cizího klíče) a databázové operace se v těchto tabulkách dají spouštět jako transakce. Transakce jsou pak většinou bezpečnější a rychlejší (28).

Tabulky typu HEAP – tabulky se neukládají fyzicky na pevný disk, existují pouze v operační paměti RAM a používají tzv. index typu hash (28).

2.15 Webová služba

Webová služba je aplikační komponenta, která je dostupná přes standardní otevřený protokol. Funguje zde princip klient (konzument) – server, kde klient klade požadavky na server a ten je zpracovává a vrací klientovi odpověď. Služba poskytuje dostatečný popis svého rozhraní a je jasně definovaný formát zpráv, kterými se komunikuje. K popisu webové služby se nejčastěji používá jazyk WSDL (Web Service Description Language), který popisuje operace a vstupní a výstupní parametry každé operace (38).

2.16 SOAP

Pro princip využívání webových služeb dostupných přes standardní protokoly se nejčastěji využívá protokolu SOAP (Simple Object Access Protocol). Tento protokol specifikuje formát zpráv, který si při komunikaci vyměňují mezi sebou klient a server. Je velmi jednoduchý a díky tomu se implementoval do mnoha platforem. Byl navržen pro internet, přesto je nezávislý na přenosovém protokolu. Dnes bývají pro přenos SOAP zpráv nejběžněji používány protokoly HTTP a HTTPS (38).

2.17 Webmin

Webmin je webové rozhraní pro správu unixových systémů. Pomocí webového prohlížeče umožňuje mimo jiné například nastavení uživatelských účtů, Apache, DNS záznamů, sdílení souborů atd. Odstraňuje potřebu ruční úpravy systémových konfiguračních souborů a umožňuje vzdálenou správu systému nebo z CLI (39).

2.18 Virtualmin

Virtualmin je modul pro Webmin. Umožňuje správu více virtuálních hostitelů prostřednictvím jednotného rozhraní. Podporuje správu a vytváření virtuálních hostitelů Apache, domén, databází MySQL, poštovních schránek a aliasů. Využívá standardní

moduly Webminu a umí pracovat s jakoukoliv systémovou konfigurací – nevyžaduje vlastní poštovní a webový server atd. (40).

Virtualmin poskytuje také vzdálené API, které umožňuje převést příkazy do volání pro příkazový řádek a následně navrátit výsledek zpět v požadovaném formátu. S jeho pomocí je tedy možné spravovat virtuální servery, uživatele, databáze a jiné funkce vzdáleně z jiných programů (41).

Veškerá vzdálená volání musí být provedena skrz CGI (common gateway interface) Virtualminu. CGI musí být veřejně dostupné a nachází se na adrese `https://hostname:10000/virtual-server/remote.cgi`, kde hostname je celý název hosta, kde se Virtualmin nachází. Každé volání je podmíněné komunikací skrz https nebo http protokol ve formátech GET nebo POST, ověřením uživatele a existencí povinného parametru „program“, který určuje, jaký úkon po API požadujeme (41).

2.19 Informační systém

Informační systém je souborem pro označení technických prostředků (informačních technologií), které poskytují funkčnost pro sběr, přenos a uchovávání dat pro požadované účely uživatelů. Sami uživatelé jsou nedílnou součástí systému, protože oni sami jsou nositeli informací. Můžeme tedy za informační systém označit cokoliv, co poskytuje uživatelům informace (například tedy tak můžeme označovat orientační cedule na chodbách, ale i systém, který řídí chod celé organizace). Podstatným výstupem informačního systému je kvalita zpracovaných informací (42).

2.19.1 Důvody k provozu informačního systému

„Každá organizace potřebuje ke svému fungování informace. K tomu aby mohla přijímat objednávky od svých zákazníků, k tomu aby mohla vyplácet mzdy nebo platy svým lidem. K tomu aby mohla vůbec fungovat. O tom o všem vede informace. A informace potřebuje sdílet, komunikovat a uchovávat. A informace potřebuje proto, aby přežila a aby mohla

zvyšovat svoji konkurenceschopnost. Kdo má lepší informace, má větší předpoklady k vyšším ziskům či menším nákladům“ (42).

K tomu, aby měly organizace dostupné správné informace ve správný čas, pomohou odpovídající technologie. Všechny tyto technologie poté pro organizaci představují informační systém a ten by měl plnit účely:

- sběru dat,
- uchovávání dat,
- přenos dat,
- zpracování dat,
- poskytování dat a informací (42).

2.19.2 Obsah informací v informačním systému organizace

Informační systém obsahuje informace a data, která jsou potřebná k provozu a fungování firem. Tyto informace se liší podle sektoru trhu, ve kterém je podnik provozován. V podstatě je možné pokrýt následující oblasti:

- řízení lidských zdrojů – informace o zaměstnancích a jejich mzdách,
- řízení financí a ekonomiky organizace – účetní informace,
- správa majetku – informace o správě majetku,
- logistika a doprava – informace o pohybu zboží, dodacích lhůtách apod.,
- prodej – informace o výrobcích, službách a zaměstnancích,
- řízení výroby a poskytování služeb,
- řízení kvality a procesů – informace o skutečných stavech a možnostech zlepšení,

- řízení projektů – informace o projektech, jejich stavu a zákaznících,
- řízení rizik – informace o možných hrozbách,
- strategická řízení – vývoj a stav strategických plánů,
- řízení bezpečnosti – informace o přístupových právech, přístupové informace,
- informatika a řízení IT – informace o dostupných aplikacích, technickém vybavení a jeho řízení,
- řízení znalostí a kontinuita organizace – různé informace důležité pro firmu (42).

2.19.3 Základní funkce informačního systému

„Informační technologie zajišťují různou funkčnost. Od základů, kdy “pouze” drží informace až po složité expertní systémy, které pomáhají plánovat. Software tedy může zajišťovat různou funkčnost, která se liší organizací od organizace, podnik od podniku“ (42).

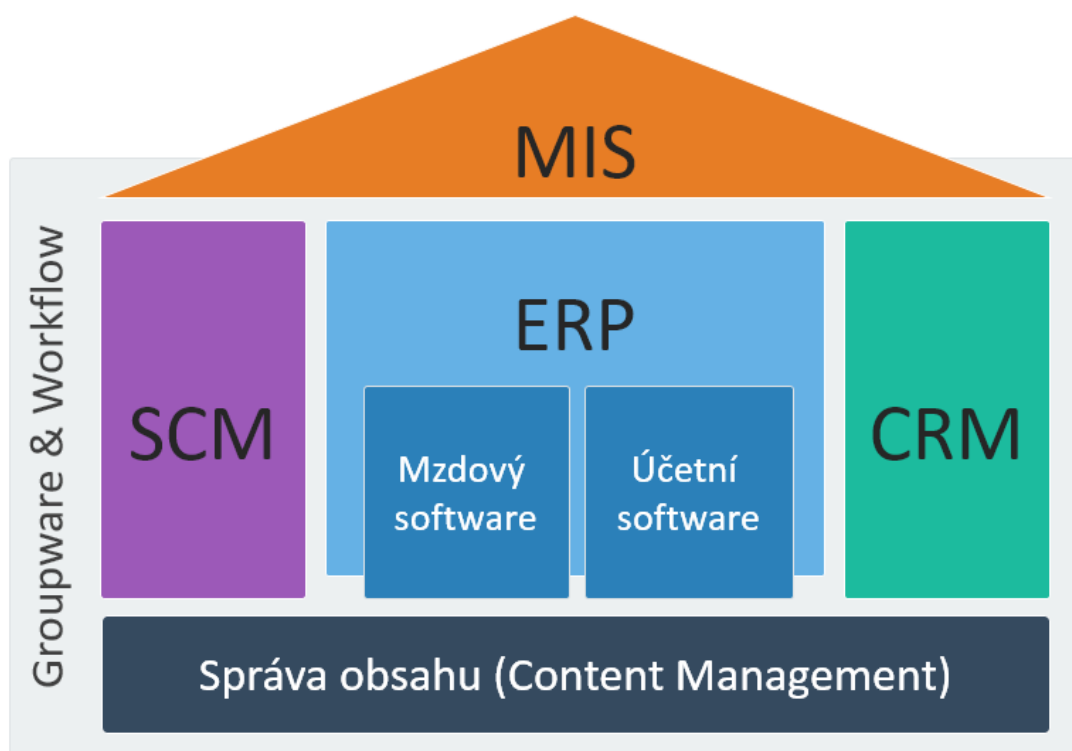
Základní funkce informačního systému se pak dají rozdělit na následující:

- evidence a uchovávání informací,
- zpracování a uchovávání transakčních informací,
- komunikace a výměna informací,
- poskytování informací pro řízení a rozhodování,
- plánování,
- simulace různých situací pro rozhodování (42).

2.19.4 Základní členění podnikových aplikací

Aplikační část informačního systému tedy netvoří pouze jedna aplikace, ale řada podnikových aplikací a nástrojů. Nejčastěji se podnikové aplikace dělí do následujících kategorií:

- ekonomické a účetní systémy,
- software pro správu lidských zdrojů (HRM),
- systémy správy podnikových zdrojů (ERP),
- systémy pro řízení výroby (APS),
- systém pro řízení vztahů se zákazníky (CRM),
- systém pro správu obsahu (ECM) (42).



Základní schéma podnikového software

Obr. 2. Základní schéma podnikového software, zdroj: (42)

2.20 Active 24, s.r.o.

Česká společnost zabývající se webhostingem a registrací domén. Nabízí sdílený i privátní hosting a managed servery na míru. Provoz serverů v datovém serveru Tower a v geograficky oddělném záložním datacentru. Všechny služby a data jsou pravidelně zálohovány. Společnost sídlí v Praze a aktuálním majitelem je severská společnost Visma Group (43).

3 ANALÝZA PROBLÉMU A SOUČASNÁ SITUACE

V této části práce se budu věnovat popisu základních informací o podniku a popisu aktuálního stavu, ve kterém se podnikový systém nachází. Na konci kapitoly také popíšu hlavní problematiku, kterou tato práce řeší.

3.1 ZÁKLADNÍ INFORMACE O SPOLEČNOSTI

Společnost XART, s.r.o. byla založena v roce 2011 jednatelem (Aleš Janoušek). Společnost se zabývá tvorbou webových prezentací, poskytováním cloudových a hostingových služeb, programováním aplikací a poradenstvím v oblasti informačních technologií. Hlavní činností firmy je vytváření webových řešení na míru a internetovým marketingem.



Obr. 3. Logo XART, s.r.o., zdroj interní podnikové materiály

3.1.1 Přehled základních informací

Datum zápisu: 14. 1. 2011

Spisová značka: 69051 C, Krajský soud v Brně

Obchodní firma: XART, s.r.o.

Sídlo: Hlavní 172/95, okres Brno-město, PSČ 62400

Identifikační číslo: 29261864

Právní forma: Společnost s ručením omezeným

Předmět podnikání: výroba, obchod a služby neuvedené v přílohách 1 až 3 živnostenského zákona

Základní kapitál: 200 000 Kč

Společníci se vkladem:

- Ing. Aleš Janoušek (Velké Meziříčí, Lesní, PSČ 594 01), vklad 170 000 Kč, obchodní podíl 85%,
- František Blažek (Jihlava, Stará cesta, PSČ 586 01), vklad 30 000 Kč, obchodní podíl 15%.

3.1.2 Organizační podmínky podniku

Společnost má pobočku ve Velkém Meziříčí, Novosady 1139/40. Celkový počet zaměstnanců na hlavní pracovní poměr je 9.

3.2 Nabízené produkty a služby společnosti

Popis primárních produktů a služeb, které společnost nabízí.

3.2.1 Internetový marketing

Aplikace nástrojů, které vedou ke zvyšování návštěvnosti webu nebo vedou k dosažení s klientem přesně definovanému cíli. Podnik poskytuje garanci plnění měřitelných cílů. Tedy zvyšování výkonnosti webu, čímž se myslí zvyšování konverzního poměru návštěvníků na webu.



Obr. 4. Ukázka garance splnění měřitelných cílů na ukázkovém anonymním projektu, zdroj: (44)

3.2.2 Tvorba www prezentací

Návrh, realizace nebo redesign webových stránek. Společnost nabízí typy webů od „záchranného“, přes firemní prezentaci až po e-shopy. Zaměřuje se hlavně na použitelnost (ziskovost) nového řešení.

Tvorba webu obsahuje	Nouzová stránka	Hezký webík	Hezká řemeslná práce	Hezké firemní stránky	Mašina na obchodní příležitosti	A jaký bude Váš web?
Grafický návrh	Ano	Ano	Ano	Ano	Ano	?
Uživatelsky přívětivá administrace		Ano	Ano	Ano	Ano	?
Administrovatelné články a aktuality		Ano	Ano	Ano	Ano	?
Katalog, E-shop, nebo jiná úprava na míru			Ano	Ano	Ano	?
Převod informací ze starého webu na nový			Ano	Ano	Ano	?
Interakce s uživatelem - pokročilý poptávkový formulář, konfigurátor				Ano	Ano	?
Vstupní analýza - Analýza klíčových slov - Analýza konkurence - ...					Ano	?
Vícejazyčný web					Ano	?

Obr. 5. Tabulka webového řešení, zdroj: (45)

Většina webových prezentací je postavená na systému xartCMS (produkční verze 3 a 6). Obě verze CMS jsou založeny na Joomla! CMS a lze využít zpětné kompatibility Joomla! na xartCMS, je tedy možné pro xartCMS používat moduly, komponenty a pluginy určené pro Joomla! CMS.

3.2.3 Programování zakázkových aplikací

Programování webových aplikací na míru dle přání zákazníka v jazyce PHP. Aplikace jsou buď součástí CMS, nebo jsou postavené na Nette Framework a jsou tedy schopné pracovat i mimo prostředí CMS.

Základní typy webových aplikací, které společnost vytváří:

- obchodní webové aplikace – B2B aplikace (propojení různých ERP systémů),
- ekonomické – management aplikace, procesní řízení Active Collab,
- benefiční – vnitrofiremní aplikace pro ohodnocování zaměstnanců,
- datové pumpy – importy a exporty dat mezi různými prostředími,
- statistické – pokročilé statistiky zobrazení stránek vhodné pro portály,
- aplikace pro marketingové účely – konfiguratory, on-line výpočty, úpravy stávajících komponent.

3.2.4 Virtualizace serverů a desktopů

Poskytování služeb webhostingu (xartCLOUD HOSTING), mailhostingu a doménových služeb (například správa DNS).

Spolehlivost xartCLOUD HOSTINGU je z pohledu garance dostupnosti SLA 99,99% (46).

3.3 Technologické vybavení

V této části budou popsány technologická vybavení podniku. Uvedena bude technologická konfigurace produkčního serveru, včetně rozdělení na logické kontejnery. Na tomto serveru se nachází v jednotlivých kontejnerech klientská řešení.

3.3.1 Konfigurace produkčního serveru

Aktuální konfigurace produkčního (veřejně dostupného) serveru:

- Hypervizor Lenovo System x3650 M5,
- 2x Intel® Xeon® CPU E5-2630 v3 2.40GHz (2x 8 jader podporující technologii Hyper Threading – procesor tedy poskytuje 32 vláken),
- 2x 4TB Toshiba MG04ACA400E - 7200rpm, SATA3, 512e, 128MB, 3,5",
- 4x 4TB WD RED (zálohy),
- 2x Intel Enterprise SSD 200GB,
- 96GB DDR4 RAM,
- 10Gb/s Intel síťová karta pro propojení mezi ostatními servery,
- OS Proxmox 4.4 (Debian),
- na discích Toshiba je souborový systém ZFS s konfigurací RAID 1,
- na discích WD RED je souborový systém ZFS s konfigurací ZRAID-1 (ekvivalent RAID 5).

Disky Toshiba jsou rozděleny na několik oddílů, část slouží jako L2arc cache pro plotnové disky 40GB, které nejsou zrcadlené, protože jejich ztráta nemá vliv na ztrátu

dat, ale při výpadku jednoho SSD se pouze sníží dostupná cache na půlku. Dále je použita „zapisovací cache“ takzvaný LOG, který je zrcadlen a má kapacitu 5GB. Zbytek SSD je vyčleněn pro samotné aplikace jako standardní ZFS pool, který je zrcadlen.

3.3.2 Produkční dělení disků na kontejnery

Produkční server se rozdělej na několik kontejnerů, z nichž každý má svoji specifickou úlohu. Rozdělení veřejně dostupných kontejnerů:

- webový – obsahuje Virtualmin pro správu hostingu webu a emailu,
- SQL – obsahuje pouze MySQL databáze, je oddělen pro snadnější monitoring a možnost lepšího škálování výkonu,
- SMTP – pro veškerou standardní poštovní komunikaci,
- SMTP Newsletter – pro odesílání newsletterů.

3.4 Podnikový informační systém

Jako podnikový informační systém je používán Active Collab verze 4.1.7, od společnosti A51. Active Collab je webový správce projektů, ale díky rozšiřujícím modulům lze rozšířit například o management, spolupráci při řešení úkolů, sledování času, fakturaci, správu financí, správa zdrojů (SVN) a integraci externích služeb (například slack, paypal, google drive, dropbox a další). Kromě toho lze programovat vlastní moduly s požadovanou funkcí. Dokáže tedy nahradit klasický informační systém v základní míře, kterou podnik od informačního systému vyžaduje.

Počet aktuálně přihlášených uživatelů není omezený.

Systém není omezen na interní prostředí, je zprovozněn na webovém serveru s přístupem odkudkoliv. Tento webový server je umístěn přímo v pobočce firmy.

Moje úkoly

Filtr: Uživatelský filtr

☐ Osobní filtr
 ☒ Zahrnout všechny projekty
 ☒ Včetně času
 ☐ Normovat součty hodin
 ☒ Zahrnout podúkoly
 ☐ Oddělit podúkoly
 Spustit

Všechny přiřazené

Štítek	(Typ) Název	Zodpovědná osoba	Projekt	Vytvořeno	Vytvořil	Termín splnění	Odhadovaný čas	Sledovaný čas	Zbývajcí čas
	(ÚKOL) #15: Merkantile	Petr H.	xartDevel	9 led 2017	Soňa K.	8 kvě 2017	4h z Programátorské práce	4,17h	0h
Nejvyšší	(ÚKOL) #84: FW: Fw: Dotaz na možnost výměny zpětných odkazů	Petr H.	xartDISPECINK	22 bře 2017	Soňa K.	8 čer 2017	3h z Programátorské práce	5h	0h
Vysoká	(ÚKOL) #306: implementace napojení systému na DPD	Petr H.	xartDevel	16 dub 2017	Petr H.	6 čec 2017	0,75h z Programátorské práce	Prázdný	0,75h
	(ÚKOL) #9: Konzultace Zásilkovna	Petr H.	xartDevel	14 úno 2017	Aleš J.	20 čer 2017	1h z Základní práce	Prázdný	1h
	(ÚKOL) #61: Rozšíření reportu denního plánu	Petr H.	xartPM	3 úno 2017	Petr H.	2 čer 2017	Prázdný	Prázdný	-
Nejnižší	(ÚKOL) #416: Odpověď #6, #9, konzultace	Petr H.	xartDevel	3 úno 2017	Petr H.	20 kvě 2017	6h z Programátorské práce	4h	2h
	(ÚKOL) #2: úkol začínající								

+ RYCHLE PŘIDAT ⚠ UPOZORNĚNÍ 11:17 ODPOLEDNE KOŠ

Obr. 6. Prostředí informačního systému, zdroj interní materiály podniku

Díky využívání technologie AJAX je práce v systému plynulá a navozuje iluzi, že celý systém ani není webovou aplikací, protože stránka není aktualizována při každé akci, ale aktualizují se pouze určité části aktuálně zobrazené stránky (klient komunikuje se serverem na pozadí webové stránky).

3.4.1 Konfigurace serveru pro provoz informačního systému

Přehled minimální, doporučené konfigurace webového serveru podle oficiální dokumentace autora informačního systému Active Collab verze 4.1.7 včetně skutečné konfigurace serveru, na kterém je v podniku provozován. Zdroj pro minimální a doporučenou konfiguraci serveru pro provoz Active Collab: (47).

Tab. 1. Konfigurace serveru pro provoz informačního systému

Typ konfigurace	Minimální konfigurace	Doporučená konfigurace	Skutečná konfigurace
Webový server	S podporou PHP 5.4	Apache s podporou PHP 5.5	Apache s podporou PHP 5.5
Podpora extenzí webového serveru	PCRE, XML, ctype, tokenizer, session, dom, phar, zlib a json	PCRE, XML, ctype, tokenizer, session, dom, phar, zlib, json, gd, mbstring, iconv, phar, zlib, IMAP a SVN	PCRE, XML, ctype, tokenizer, session, dom, phar, zlib, json, gd, mbstring, iconv, phar, zlib, IMAP a SVN
Databáze	MySQL	MySQL 5 databáze s úložištěm typu InnoDB	MySQL 5 databáze s úložištěm typu InnoDB
Limit paměti serveru	nespecifikováno	minimálně 128 MB	256 MB

3.4.2 Doporučené požadavky na klienta pro využívání Active Collab verze 4.1.7

Minimální požadavky na klienta (webový prohlížeč) pro práci s informačním systémem podniku podle oficiální dokumentace autora Active Collab (47):

- internetový prohlížeč Internet Explorer 9 a vyšší,
- poslední verze internetového prohlížeče Google Chrome,
- poslední verze internetového prohlížeče Mozilla Firefox,
- poslední verze internetového prohlížeče Apple Safari.

Systém je optimalizovaný pro rozlišení obrazovky 1280x800 px (desktopová verze), ale aplikace je responzivní a rozeznává různé způsoby zobrazení, proto dochází ke správnému zobrazení i na mobilních zařízeních.

3.4.3 Využívané moduly systému

Seznam modulů, které rozšiřují informační systém a které podnik využívá.

- Export projektu – export projektu jako statickou webovou stránku (vytváření náhledů a přístupů klientů k úkolům k projektu bez možnosti zasahovat),
- Seznamy úkolů – přidávání seznamů úkolů k projektům,
- Dokumenty – správa dokumentů,
- Soubory – správa souborů projektů,
- Zápisy z jednání – zápisník projektu pro přidávání záznamů z jednání s klienty,
- Úkoly – správa úkolů v projektech,
- Sledování času a výdajů – sledování stráveného času a vynaložených výdajů v úkolech projektu,
- Fakturace – správa faktur,
- Zdroj – propojení s verzovacím systémem serveru (SVN),
- Opakované úkoly – možnost vytvářet a spravovat periodicky se opakující úkoly,
- Hledat Plus – zrychluje vyhledávání ve všech modulech systému,
- Úkoly Plus – vylepšení správy úkolů,
- Časové reporty Plus – generování vlastních časových reportů sumarizovaných podle různých parametrů,

- Easy CRM – evidence potenciálních zákazníků pomocí rozšířených profilů, převod na skutečné zákazníky, ukládání historie komunikace se zákazníky,
- Widget Plus – personifikace pohledových karet,
- Stopky Plus – vylepšené sledování času stráveného na projektu,
- Komunikace – komunikace se členy týmu a klienty,
- Hesla – bezpečná správa hesel,
- Šablony úkolů – vytváření úkolů pomocí předpřipravených šablon,
- Planium – zobrazování a správa úkolů v projektu pomocí Kanbanového panelu.

Z využívaných modulů systému lze tedy určit, že informační systém pokrývá (alespoň z části) následující podnikové procesy:

- řízení projektů,
- řízení kvality a procesů,
- řízení financí a ekonomiky organizace.

3.4.4 Práva v prostředí systému

Uživatelská oprávnění pro práci v systému se definují na třech základních úrovních. Těmito úrovněmi jsou:

- systémové role,
- vlastní uživatelská oprávnění jednotlivých modulů,
- role v projektu.

3.4.5 Povolené role v systému

Uživatelům jsou přiřazovány jednotlivé role (skupiny oprávnění), které mají nastavené sbírky oprávnění. Aktuální jsou:

- administrátor – jsou mu udělena všechna oprávnění, včetně oprávnění přihlášení pod cizím účtem bez znalosti přihlašovacích údajů uživatele,
- manažer – možnost spravovat zaměstnance, projekty a finance, v některých modulech vlastní oprávnění na stejné úrovni jako administrátor,
- zaměstnanec – smí vytvářet úkoly v projektech, evidovat hodiny k úkolům, a vykonávat běžnou pracovní činnost v modulech, ke kterým má udělený přístup,
- subkontraktor – externí zaměstnanec, smí evidovat hodiny k úkolům, k většině modulům nemá přístup (moduly samy nedovolují konfigurovat oprávnění přístupu k tak nízké úrovni oprávnění),
- klient – pouze přístup k klientova projektu, zakázána správa projektu, možnost komunikace skrz komentáře k úkolům,
- omezený klient – pouze přístup do klientova projektu, zakázána správa projektu.

Vlastní uživatelská oprávnění

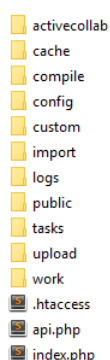
Tato oprávnění umožňují nastavit uživateli s určitou systémovou rolí individuální oprávnění, toto oprávnění tedy umožňuje specifikovat speciální oprávnění, které mohou moduly zohledňovat při kontrole oprávnění k určitým akcím.

Seznam těchto oprávnění pro konkrétní role v systému rozšiřují samy moduly. Mezi tato oprávnění pak patří například: „Používání časových reportů“, „Správa koše“ a další.

Role v projektu

V projektu je možné určit vedoucího projektu (kdo projekt řídí). Projekty jsou pak složeny mimo jiné i z úkolů, ke kterým lze přiřadit zodpovědnou osobu (kdo zodpovídá za splnění úkolu). Těmto úkolům je následně možné přiřadit i podúkoly (jednotlivé kroky při realizaci úkolu) a těm je také možné přiřadit zodpovědnou osobu.

3.4.6 Páteří adresářová struktura souborů systému



Obr. 7. Páteří adresářová struktura souborů systému, zdroj vlastní zpracování

Popis funkce adresářů kořenové adresářové struktury informačního systému:

- activecollab/ – jádro systému, framework, systémové moduly,
- cache/ – odkládání dočasných souborů,
- compile/ – úložiště překompilovaných šablon PHP,
- config/ – konfigurační soubory (například připojovací údaje k databázi),
- custom/ – rozšiřující moduly třetích stran,
- import/ – úložiště dočasných souborů sloužící pro import dat,
- logs/ – chyby, varování a upozornění systému,

- public/ – veřejně dostupné spouště systému,
- tasks/ – skripty určené pro spouštění plánovanými úlohami,
- upload/ – ukládání nahraných souborů,
- work/ – pracovní soubory modulů.

3.4.7 Páteří aplikační struktura systému

Informační systém využívá architekturu Model-View-Controller (MVC) rozdělenou na logické celky, které označuje jako framework a moduly. Celá aplikace systému je postavena na frameworku nazývaný „Angie“, což je blíže nespecifikované označení frameworku přímo od tvůrců systému. Funkčnost frameworku je obecná a není ji třeba nijak měnit.

Hlavní (jaderný) modul se nazývá „systém“, ten se celý stará o vybudování aplikace systému a integraci dalších externích modulů (rozšíření). Pokud se tedy bavíme o modulu faktur či správu projektů, jsou to právě reprezentanti rozšiřujících modulů hlavního modulu systém.

Framework i modul mají podobnou vnitřní strukturu, kde se mohou objevit následující soubory a adresáře:

- init.php – povinný soubor, inicializační skript rozšíření, nastavuje identifikační konstanty,
- (název rozšíření)Framework.class.php nebo (název modulu)Module.class.php – povinný soubor, definuje směrování, reakce na systémové události apod.,
- assets/ – adresář statických souborů,
- controllers/ – MVC kontroléry rozšíření,

- `handlers/` – adresář se skripty s metodami, které jsou pomocí hlavního souboru rozšíření zaregistrovány jako reakce na systémové události,
- `helpers/` – dodatečné skripty pro použití v šablonách Smarty,
- `notifications/` – adresář se skripty obsahující třídy pro systémová upozornění a šablony pro e-mailová upozornění,
- `models/` – MVC modely rozšíření,
- `resources/dictionary.clientside.php` – seznam řetězců k překladu textů použitých v JavaScriptu,
- `resources/dictionary.serverside.php` – seznam řetězců k překladu textů použitých v PHP,
- `views/default/` – šablony pro desktopové rozhraní,
- `views/phone/` – šablony pro mobilní rozhraní,
- `views/printer/` – šablony pro tiskové rozhraní,
- `widgets/` – adresář pro kolekce JavaScriptových, kaskádových a případně PHP souborů a obrázků, které lze naimportovat do šablony a rozšířit tak její funkčnost.

3.4.8 Překlady rozšíření

Pro překládání textů se používá překladový systém, který mapuje výchozí anglické řetězce na lokalizované řetězce. Každý řetězec, který se má zobrazit uživateli, by měl být napsán v anglickém jazyce a odeslán do překladové funkce. Uživateli se poté zobrazí až výstup z této funkce. Díky této funkčnosti lze skrz administrátorské rozhraní překládat systém dle potřeby do různých jazyků.

V PHP lze v systému využít statická metoda „`lang()`“ třídy „Globalization“. V šablonovacím systému Smarty lze využít blok překladu „`{lang}...{/lang}`“. Pro překlad

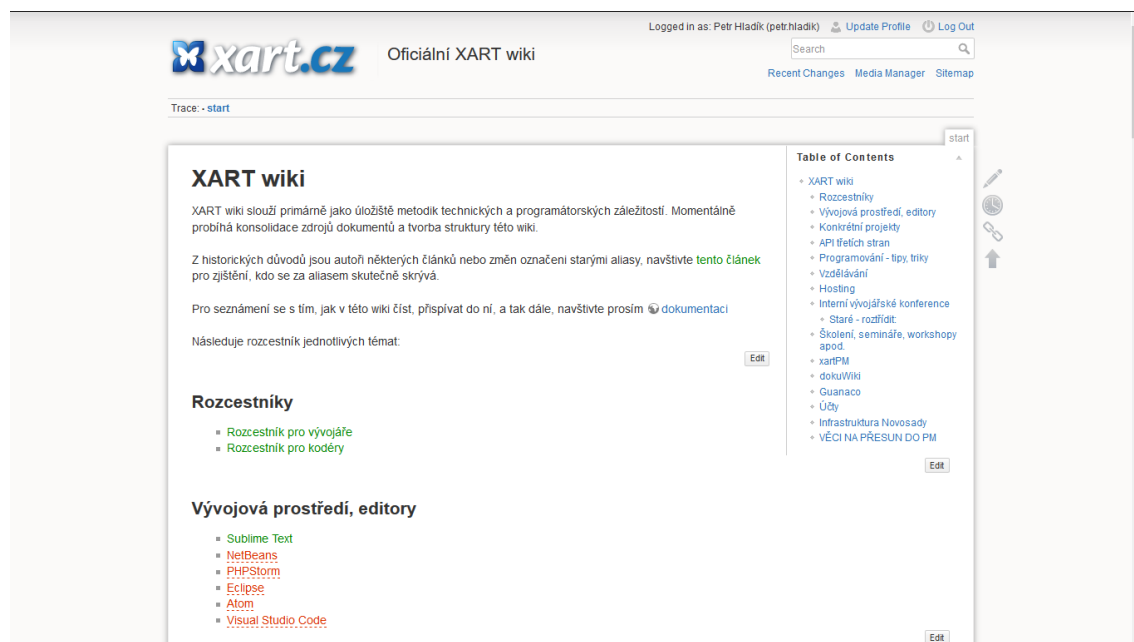
v JavaScriptu se pak využívá funkce „App.lang()“. U všech těchto funkcí se očekává vstupní parametr text k přeložení (v originálním anglickém jazyce).

Všechny texty k překladu je nutné uvést v souborech rozšíření „resources/dictionary.clientside.php“ a „resources/dictionary.serverside.php“. Všechny tyto texty jsou uloženy, pro rychlejší práci, do databázové cache aplikace. Vyvolat opětovné načtení překladových souborů je možné přes administrátorské rozhraní systému.

3.5 XART Wiki

Ve firmě je mimo informačního systému také zavedena vnitropodniková wikipedie pro shromažďování firemního know-how. Sem se umísťují poznatky, postupy, tipy a konvence pro řešení různých programátorských řešení a všeobecné informace. Vnitropodniková wikipedie je nekomerční open-source aplikace DokuWiki.

Oficiální stránky DokuWiki – <https://www.dokuwiki.org> (48).



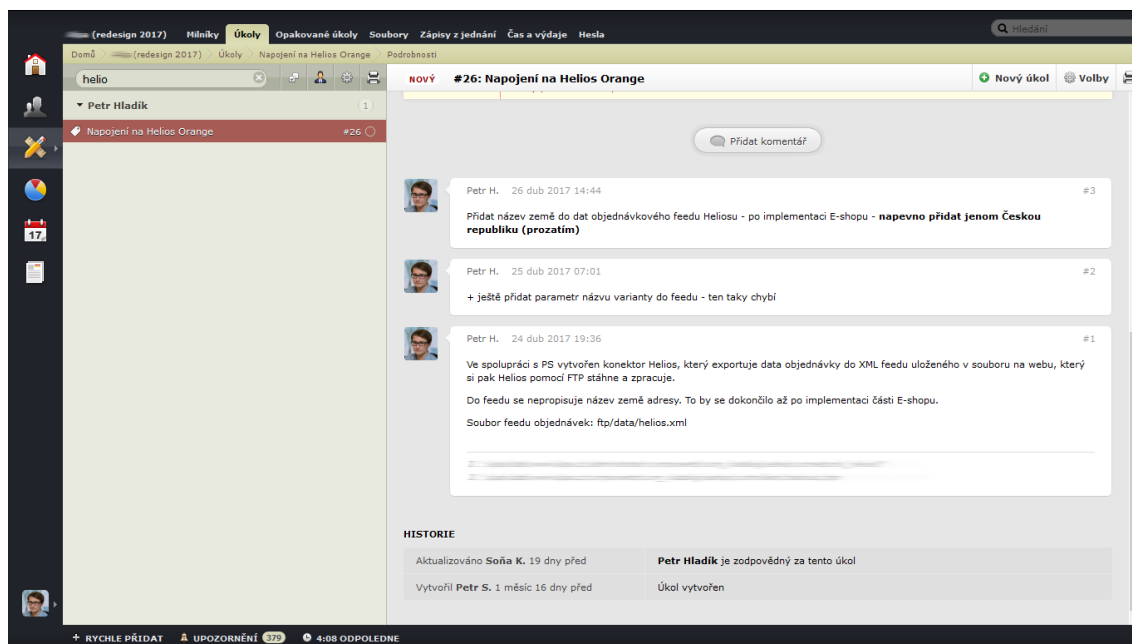
Obr. 8. XART Wiki, zdroj interní podnikové materiály

3.6 Specifikace problematiky

Hlavní problematikou, kterou se práce zabývá, je aktuální způsob ukládání informací o klientských webech. O nich je vhodné shromažďovat následující informace:

- konfigurace hostingu (domény),
- konfigurace DNS záznamů,
- typ a verze CMS,
- typy a verze rozšíření CMS,
- databáze.

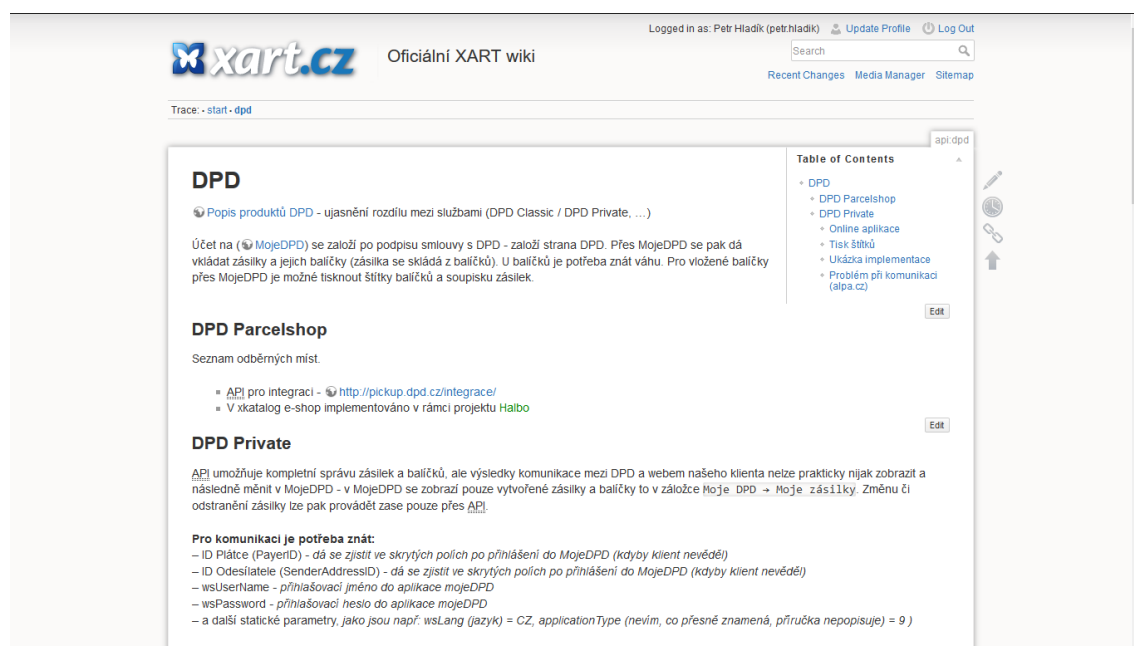
Z největší části jsou tyto informace shromažďovány do informačního systému v podobě komentářů k jednotlivým úkolům zařazených v projektech, které se týkají konkrétního klientského webu.



Obr. 9. Zaznamenávání informací pomocí komentářů, zdroj vlastní zpracování

Ukládání informací o konkrétních řešení na klientských webech pomocí zápisu úprav do komentáře v úkolu v projektu, zdroj vlastní zpracování.

Některé další informace (např. souhrn způsobu implementace aktuálního většího rozšíření webu) jsou pak zaneseny do XART Wiki nebo v případě starších webů sepsány do dokumentů ve správci projektů.



Obr. 10. Zaznamenávání informací pomocí XART Wiki, zdroj vlastní zpracování

Pokud je tedy vyžadován náhled aktuálního stavu klientského webu s přehledem aktuálních rozšíření, musí se ručně prohledat administrace CMS na klientském webu (pokud klientský web obsahuje web s CMS), dohledat komentáře popisující stav webu v úkolech projektů zabývajících se implementací tohoto rozšíření na tomto konkrétním webu a prohledat XART Wiki, zda obsahuje záznam o této úpravě. Díky tomu dochází k nekonzistentnosti dat, např. rozšíření je z webu odebráno a tato informace se pak již zpětně nezanese do XART Wiki.

Dohledávání aktuálních informací o klientově webu je díky všem uvedeným aspektům zdouhavé a dochází k přehlížení nebo nezjištění některých informací způsobených nekonzistentností nebo neaktuálností uložených informací na různých místech. Právě

díky nekonzistenci uložených informací na více místech v různých aplikacích není tento způsob uchovávání informací zcela vhodný. Bylo by tedy vhodné navrhnout a implementovat modul informačního systému, který bude sám automaticky detekovat aktuální stav klientského webu (i s jeho rozšířeními, DNS a databázemi) a následně je v přehledné formě interpretovat uživateli informačního systému.

Díky tomuto řešení bude snadné dohledat informace o konkrétním webu na jednom místě v přehledné formě a díky automatizaci bude také zaručena aktuálnost interpretovaných dat.

4 Návrh řešení

V této kapitole se budu zaměřovat na návrhovou a implementační část vytvoření modulu pro podnikový informační systém. Modul bude sestaven tak, aby vyhověl požadavkům zadavatele a na základě procesů identifikovaných v předešlé kapitole přinesl zaměstnancům ve firmě přínos ve formě zefektivnění vyhledávání informací o konkrétním webu, tedy snížení potřebného času pro dohledání informací a zaručení aktuálnosti interpretovaných dat.

4.1 Základní požadavky na modul

Společně se zaměstnanci společnosti jsme sjednali požadavky na základní vlastnosti modulu:

- periodická automatická aktualizace všech informací,
- možnost manuálního vyvolání aktualizace dat,
- dynamicky sestavovaný seznam domén na serveru,
- aktualizace konfigurace domén,
- aktualizace konfigurace plánu domén,
- aktualizace seznamu aktivních rozšíření nacházejících se na doménách,
- aktualizace DNS záznamů domén,
- ukládání historie změn údajů domén,
- přidružení domén ke konkrétnímu projektu,
- minimálně dvě úrovně oprávnění (uživatel, správce).

Hlavní požadovanou vlastností modulu je rychlé poskytování informací o konkrétních doménách, tudíž informace se musejí aktualizovat automaticky pomocí plánované úlohy

opakované v periodických cyklech nebo manuálním vyžádáním uživatele a po vyžádání zobrazení informací o konkrétní doméně musí dojít již pouze k zobrazení stažených informací. Do jisté míry se tím tedy zvýší šance zobrazení neaktuálních dat, ale v poměru k rychlosti načítání zobrazovaných informací je tato míra zanedbatelná.

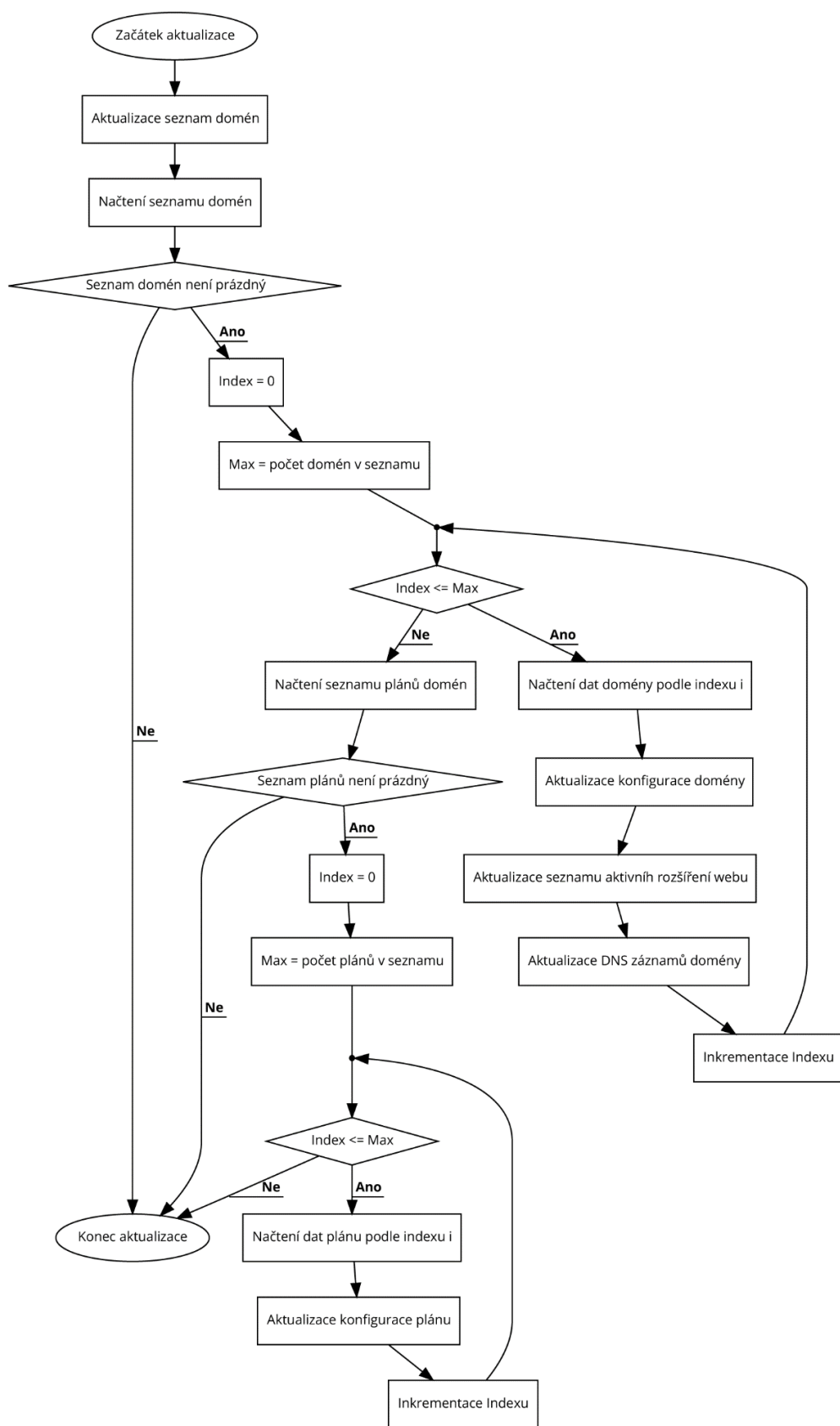
Modul musí být dále schopen získávat data z různých zdrojů (Virtualmin, Active24). Virtualmin umožňuje komunikaci přes vzdálené API (skrze CGI, odpovědi ve formátech JSON nebo XML), Active24 poskytuje API realizované přes SOAP.

4.2 Postup při aktualizaci dat

Detailněji se zaměříme na postup při aktualizaci dat domén na serveru. Ty jsou realizovány několika kroky, při nichž se komunikuje s několika vzdálenými zdroji. Tyto kroky primárně dle požadavků zadavatele jsou:

- Aktualizace seznamu domén (Virtualmin API)
- Aktualizace konfigurace domén (Virtualmin API)
- Aktualizace rozšíření webů (SQL dotazy na databáze webu a detekce souborů v úložišti webu)
- Aktualizace DNS záznamů domén (Active 24, s.r.o. API)
- Aktualizace konfigurací plánů domén (Virtualmin API)

Proces celé aktualizace tedy bude nutné řešit pomocí specifických procesů, jejichž návaznost je možné znázornit pomocí vývojového diagramu.



Obr. 11. Vývojový diagram procesu aktualizace dat domén, zdroj vlastní zpracování

4.2.1 Aktualizace seznamu domén

Nejprve je nutné detekovat seznam domén, které jsou na produkčním serveru stále aktivní. Seznam existujících domén poskytuje Virtualmin. Tyto informace získáme vyvoláním příkazu „list-domains“ bez dodatkového parametru „domain“. V tomto případě jako odpověď vykonaného příkazu navrací seznam všech existujících domén v jeho správě. Jako další dodatkové parametry uvedeme ve volání parametr „multiline“, kterým Virtualminu oznámíme, že očekáváme víceřádkový výstup (místo obvyklého jednořádkového) a parametr „json=1“, kterým určíme, že výstup z vykonané funkce má převést do formátu JSON.

V PHP tyto data získáme jednoduše vyvoláním příkazu `shell_exec` s použitím příkazu systému WGET a příslušnými parametry, získání dat z Virtualminu pak bude vypadat následovně:

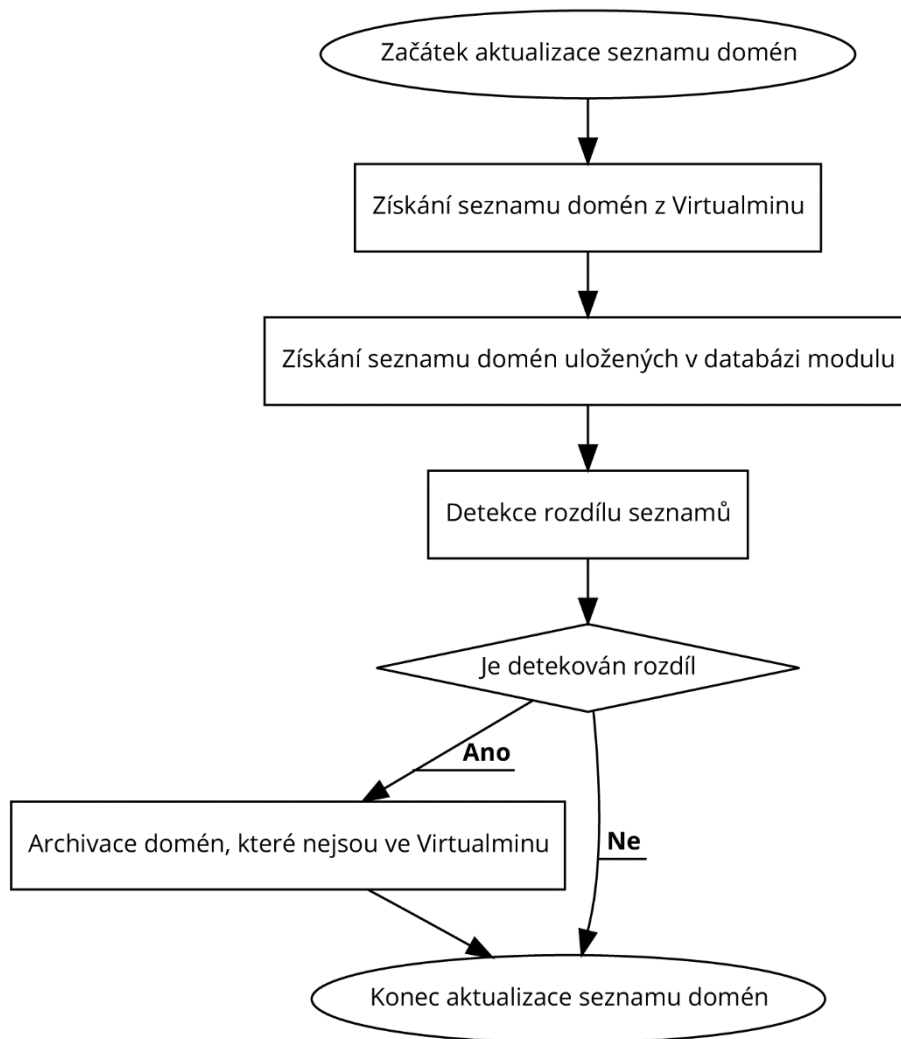
```
// Aktualizace seznamu domén
$user = '...';
$pass = '...';
$cgi = 'https://185.87.100.100:10000/virtual-server/remote.cgi';
$url = $cgi . '?program=list-domains&json=1';
$cmd = "wget -O - --quiet --http-user=" . $user . " --http-passwd=" . $pass . " --no-check-certificate " . $url . "";
$data = json_decode(shell_exec($cmd));
```

Obr. 12. Aktualizace seznamu domén z Virtualminu, zdroj vlastní zpracování

Následně je zapotřebí získaná data zpracovat a aktualizovat je v databázi.

Virtualmin vrací informace o všech doménách v poli výsledků. Informace o každé doméně se skládají z asociovaného pole, kde „name“ je název domény a „values“ je pole konfigurací domény. Seznam všech uvedených domén ve výsledku volání této funkce je seznam všech existujících domén na serveru.

Domény, které v tomto seznamu nejsou uvedeny, byly odstraněny nebo přesunuty na jiný server. Tento seznam je tedy nutné porovnat se seznamem domén uloženým v databázi modulu a případné domény navíc (které nejsou v seznamu navraceného z Virtualminu) zarchivovat.



Obr. 13. Vývojový diagram postupu při aktualizaci seznamu domén, zdroj vlastní zpracování

4.2.2 Aktualizace konfigurace domén

Pro aktualizace konfigurace domén využijeme podobné postupu, jako při aktualizaci seznamu domén. Využijeme volání na CGI se stejnými parametry, jenom přidáme parametr „domain=<název domény>“. Čímž získáme pouze informace o konfiguraci konkrétní domény.

Kompletní volání v PHP pak může vypadat následovně:

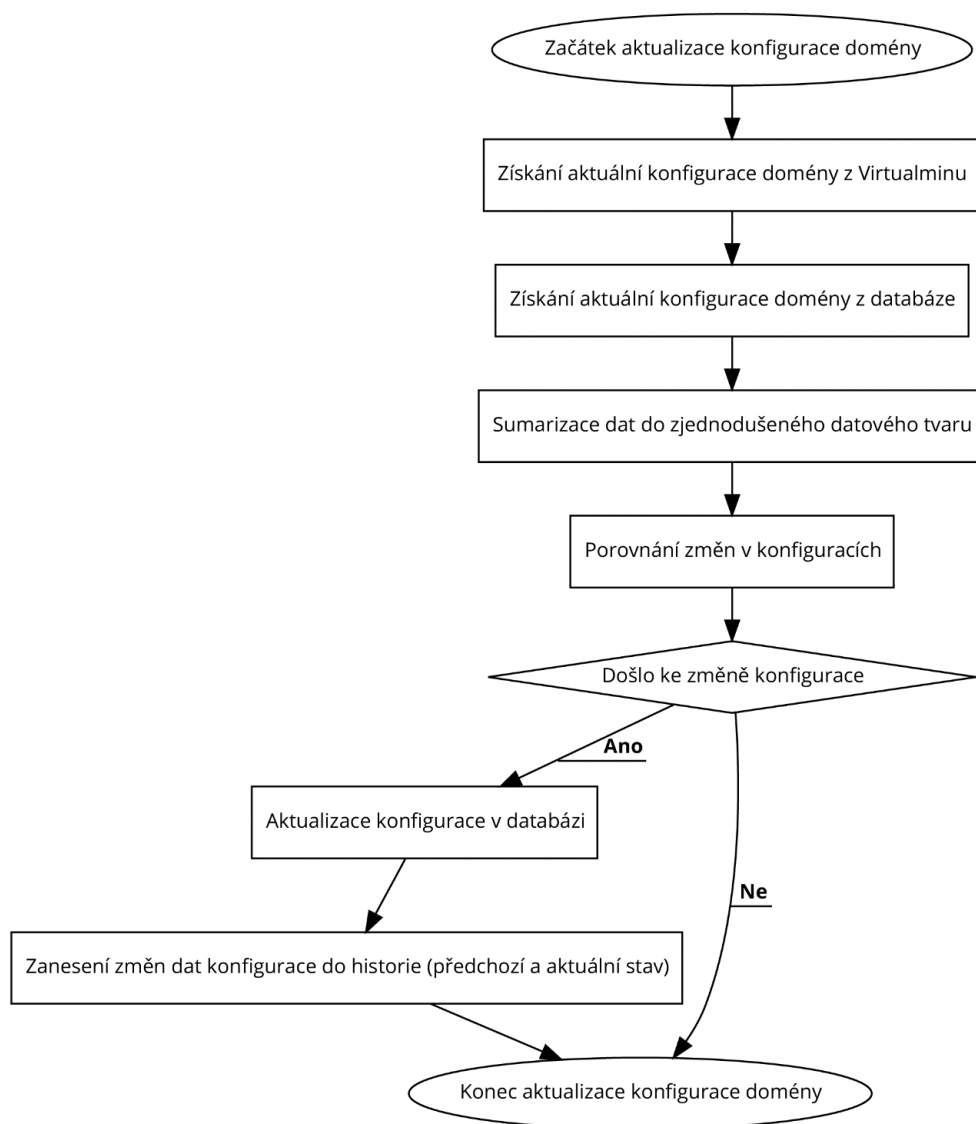
```
// Aktualizace konfigurace domén
$user = '...';
$pass = '...';
$cgi = 'https://185.87.100.100:10000/virtual-server/remote.cgi';
$url = $cgi . '?program=list-domains&json=1&domain=xart.cz';
$cmd = "wget -O - --quiet --http-user=" . $user . " --http-passwd=" . $pass . " --no-check-certificate " . $url . "";
$data = json_decode(shell_exec($cmd));
```

Obr. 14. Aktualizace konfigurace domén z Virtualminu, zdroj vlastní zpracování

Navracené informace mají stejnou strukturu jako data navracená při aktualizaci seznamu domén, „name“ je název domény a „values“ je pole konfigurací domény.

Pole konfigurací je asociované, kde klíč je název parametru konfigurace a hodnota je pole hodnot konfigurace. Pole hodnot konfigurace obsahuje právě jednu hodnotu, můžeme proto zjednodušit strukturu pole na tvar klíč – hodnota (název parametru konfigurace – konfigurace).

Navracená data konfigurace domény je potřeba následně zkontrolovat s konfigurací aktuálně uloženou v databázi, porovnat mezi nimi změny, zaktualizovat v databázi a uvést do historie změn.



Obr. 15. Vývojový diagram postupu při aktualizaci konfigurace domény, zdroj vlastní zpracování

4.2.3 Aktualizace rozšíření webů

Rozšířením webů se myslí jednotlivá CMS, jejich moduly, pluginy, komponenty a řešení na míru. Není potřeba například detekovat všechny rozšíření CMS (o některých nemá význam uchovávat informace, protože se vyskytují na všech webech s tímto CMS), ale je vhodné detekovat rozšíření, která jsou na míru nebo nespádají do „standardní“ služby pro klienty.

Detekce aktivního rozšíření může probíhat na základě dvou primárních testů:

- test existence souborů v úložišti webu a případně test obsahu těchto souborů (vyhledávání a porovnávání řetězců),
- test za použití SQL dotazů na databáze webu.

Díky staženým konfiguračním informacím z Virtualminu víme přesně, kde se adresář webu na serveru nachází a jaké databáze doména využívá. Můžeme tedy pomocí PHP detekovat existenci souborů. Databázový objekt systému podporuje navazování a udržování připojení k více databázím současně, díky tomu lze využít stejných metod pro čtení záznamů v databázích webů, jako je použito v celém informačním systému pro manipulaci s databází systému.

Pro přístup k databázím webu byl vytvořen speciální účet, který má povolené pouze oprávnění pro čtení záznamů. Skrz tento účet jsou následné testy prováděny.

Pro přístup k souborům webu bylo vytvořeno speciální spojení serverů realizováno protokolem SMB. Do konkrétního adresáře v kořenovém adresáři serveru, na kterém je umístěný informační systém, byl připojen produkční webový server a to tak, že odkazuje do adresáře, ve kterém se nachází adresáře webů. Informační systém tak má skrz PHP přístup do všech souborů webů umístěných na produkčním serveru.

Jednotlivé testy se organizují do struktur, kde je možné určit potomky testu. Kontrolér testů pak netestuje všechny dostupné testy, ale postupuje na základě těchto vazeb. Prakticky tedy začne detekovat pouze CMS a až podle aktivního CMS dále provádí testy pouze určité skupiny modulů, komponent, pluginů a řešení na míru. Tyto testy mohou mít dále své potomky atd. Důvodem tohoto řešení je optimalizace celého kroku detekce – neprovádět testy, u kterých je nereálná šance pozitivního výsledku.

```

/**
 * Loads all active extensions on domain
 *
 * @param array $extensions List of available extensions
 * @param array &$active List of active extensions - for recursive search
 *
 * @return array active extensions
 */
private function loadDomainActiveExtensionsList(array $extensions, array &$active)
{
    if (empty($extensions))
    {
        return;
    }
    foreach ($extensions as $class)
    {
        $extension = new $class();
        $extension->setDomainPath( $this->domain->html_directory );

        // db connections
        if ($extension->isEnabledCheckingDB())
        {
            $db_connections = $this->getServerDBsConnections();
            $extension->setDomainDBsConnections($db_connections);
        }

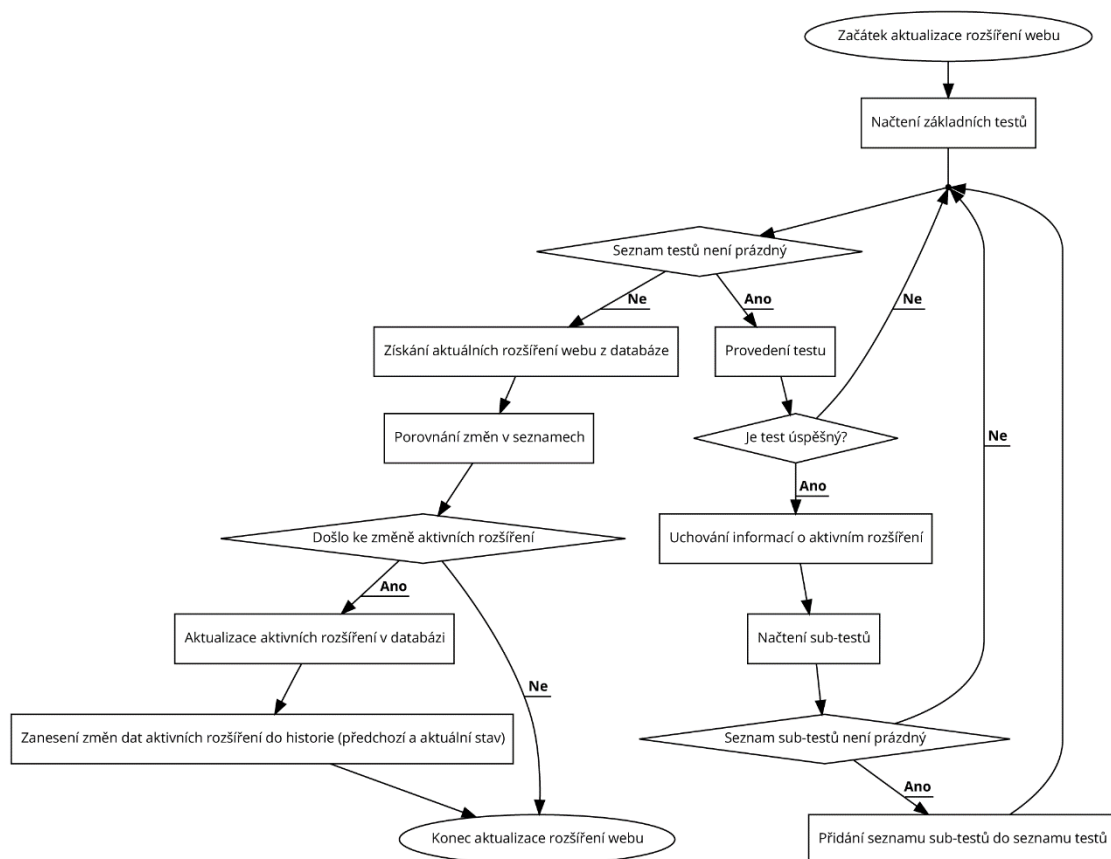
        // is active extension?
        if ($extension->isActive())
        {
            $active[] = $extension;

            // subextensions
            $subextensions = $extension->getSubExtensions();
            $this->loadDomainActiveExtensionsList($subextensions, $active);
        }
    }
}

```

Obr. 16. Metoda pro aktualizaci aktivních rozšíření domény s dynamickou kontrolou potomků, zdroj vlastní zpracování

Po dokončení testů (a sestavení seznamu aktivních rozšíření) se porovnají získaná data s těmi, co jsou uloženy v databázi a uloží se aktualizace těchto informací.



Obr. 17. Vývojový diagram postupu při aktualizaci seznamu aktivních rozšíření webu, zdroj vlastní zpracování

4.2.4 Aktualizace DNS záznamů domén

Společnost XART, s.r.o. využívá jako registrátora doménových adres výhradně společnost Active 24, s.r.o., a proto je nutné při detekci konfigurací DNS záznamů domén komunikovat s touto společností. Společnost Active 24, s.r.o. poskytuje SOAP API pro kompletní správu DNS záznamů.

Pro získání informací o konfiguracích DNS záznamů využijeme knihovnu NuSOAP (jako náhradu za SOAP extenzi, která je na serveru zakázána). Popis webové služby se nachází na adrese <https://centrum.active24.cz/services/a24PartnerService?wsdl>. Před voláním samotné funkce pro získání DNS záznamů je nutné vyvolat funkci „login“ s parametry pro ověření uživatele (jméno a heslo klienta Active 24, s.r.o.). Následně již můžeme vyvolat funkci „getDnsRecords“ webové služby pro získání DNS záznamů. Tato funkce

očekává jediný parametr a tím je název domény, ke které vyžadujeme získání DNS záznamů.

Funkce navrací pole DNS záznamů různých typů. Každý DNS záznam pak obsahuje vlastnosti:

- from,
- dns_id,
- name,
- to,
- ttl,
- type.

A podle typu záznamu i svoje primární vlastnosti, například:

- mailserver – u MX záznamu,
- alias – u CNAME záznamu,
- text – u TXT záznamu,
- ip – u A a AAAA záznamu,
- nameServer – u NS záznamu.

Mezi seznam DNS záznamů je i SOA DNS záznam, který obsahuje více parametrů, než ostatní DNS záznamy. Tyto parametry jsou:

- email,
- expire,
- dns_id,

- minimal_ttl,
- name,
- name_server,
- refresh,
- retry,
- seriál,
- to,
- ttl.

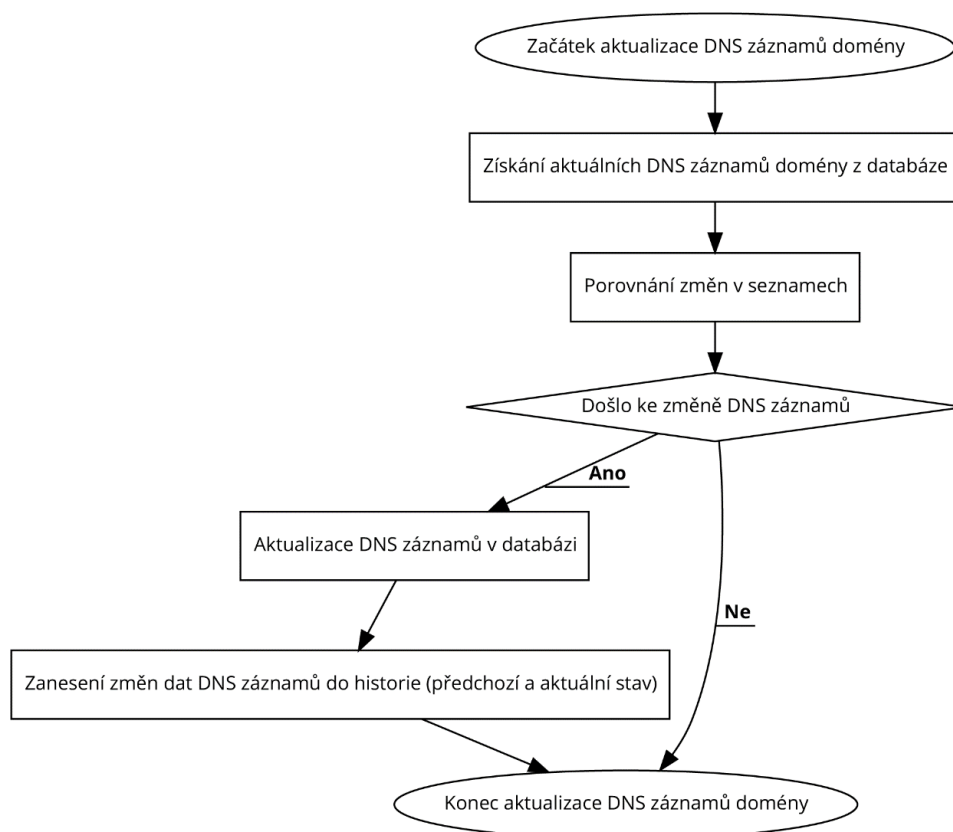
Díky skutečnosti, že parametrů obsahuje SOA záznam více a liší se od standardních DNS záznamů, je vhodné je ukládat do samostatné databázové tabulky.

Po získání konfigurací DNS záznamů domény je nutné ji porovnat s konfigurací uloženou v databázi a zaktualizovat informace.

Kompletní volání v PHP pak může vypadat následovně:

```
// Aktualizace DNS záznamů domén
$client = new nusoap_client('https://centrum.active24.cz/services/a24PartnerService?wsdl');
$data   = $client->call('login',array('...', '...'));
$data   = $client->call('getDnsRecords',array('xart.cz'));
```

Obr. 18. Aktualizace DNS záznamů domén od Active 24, s.r.o., zdroj vlastní zpracování



Obr. 19. Vývojový diagram postupu při aktualizaci DNS záznamů domény, zdroj vlastní zpracování

4.2.5 Aktualizace konfigurací plánů domén

Po aktualizaci hlavních informací o doménách je nutné také aktualizovat data konfigurací všech plánů, které mají domény použité. Plán ve Virtualminu definuje omezení kvót pro využívání systému doménou. Informace o konfiguraci plánu získáme vyvoláním příkazu „list-plan“ s parametry „name=<název plánu>“, „multiline“ (víceřádkový výstup) a „json=1“ (pro výstup v JSON formátu).

V PHP tyto data získáme jednoduše opět vyvoláním příkazu `shell_exec` s použitím systémového příkazu `WGET` a příslušnými parametry.

```
// Aktualizace konfigurací plánů domén
$user = '...';
$pass = '...';
$plan = 'Default plan';
$cgi = 'https://185.87.100.100:10000/virtual-server/remote.cgi';
$url = $cgi . '?program=list-plans&json=1&multiline&name=' . $plan;
$cmd = "wget -O - --quiet --http-user=" . $user . " --http-passwd=" . $pass . " --no-check-certificate '" . $url . "'";
$data = json_decode(shell_exec($cmd));
```

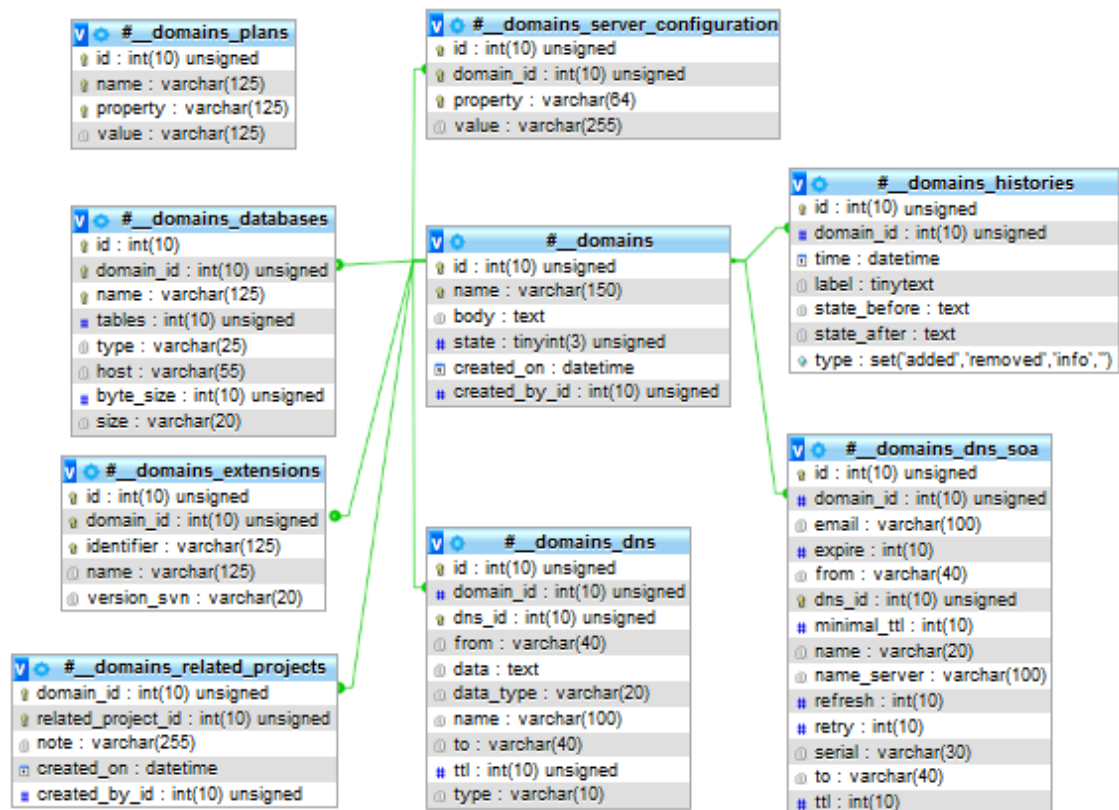
Obr. 20. Aktualizace konfigurací plánů domén z Virtualminu, zdroj vlastní zpracování

Je tedy nutné sestavit seznam všech plánů použitých v konfiguracích domén, získat informace o jejich konfiguracích a tato data aktualizovat v databázi.

4.3 Datový model

Po stanovení kroků nutných k aktualizaci informací o webech a doménách a po analýze získaných dat z externích systémů jsme schopni navrhnout datový model a vymodelovat databázové relace včetně vzájemných vazeb potřebných pro uchovávání získaných informací. Do návrhu nám vstupují následující relace:

- domény,
- konfigurace domén,
- databáze,
- DNS záznamy,
- SOA DNS záznamy,
- konfigurace plánů,
- rozšíření domén,
- historie změn,
- vazby na projekty.



Obr. 21. Relace databázových relací, zdroj vlastní zpracování

Nyní si rozklíčujeme a popíšeme relace datového modelu včetně ukázky n-tic uložených v relacích.

4.3.1 Relace „Domény“ (#__domains)

Představuje základní databázovou tabulku modulu. Obsahuje ty nejzákladnější informace o doméně a poskytuje datovému modelu základ pro relační připojení ostatních databázových relací modulu.

Tab. 2. Struktura relace „Domény“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
name	Varchar (150)	Název
body	Text	HTML popis
state	Unsigned tinyint (3)	Stav publikace
created_on	Datetime	Datum a čas vytvoření
created_by_id	Unsigned integer (10)	ID uživatele, který vytvořil

Jako primární klíč je zvolen atribut „id“, dále je v relaci definován unikátní klíč na atributu „name“ z toho důvodu, že název domény se v seznamu domén může objevovat pouze jednou.

Příkladem n-tice relace v relaci „Domény“ může být záznam:

„1;dobawebowa.cz;<p>První testovací doména<p>;1;2017-01-01 12:00:00;13“

4.3.2 Relace „Konfigurace domény“ (#__domains_server_configuration)

Slouží k ukládání hodnot konfigurací domény, které jsme schopni získat s Virtualminu.

Tab. 3. Struktura relace „Konfigurace domény“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor

domain_id	Unsigned integer (10)	ID domény
property	Varchar (64)	Název vlastnosti
value	Varchar (255)	Hodnota vlastnosti

Jako primární klíč je zvolen atribut „id“. Zároveň byl vytvořen unikátní index atributy „domain_id“ a „property“ (konkrétní parametr konfigurace se může objevit na doméně pouze jednou).

Příkladem n-tice relace v relaci „Konfigurace domény“ může být záznam:

„106443;55;php_execution_mode;fcgid“

4.3.3 Relace „Databáze domény“ (#__domains_databases)

Slouží k ukládání základních informací o databázích domén. Zdrojem těchto informací je opět Virtualmin.

Tab. 4. Struktura relace „Databáze domény“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
domain_id	Unsigned integer (10)	ID domény
name	Varchar (125)	Název databáze
tables	Unsigned integer (10)	Počet tabulek databáze
type	Varchar (25)	Typ databáze

host	Varchar (55)	Host databáze (IP)
byte_size	Unsigned integer (10)	Velikost [B]
size	Varchar (20)	Velikost [MB]

Primárním klíčem je zvolen atribut „id“ a byl zvolen unikátní index na attributech „domain_id“ a „name“.

Příkladem n-tice relace v relaci „Databáze domény“ může být záznam:

„19;dobawebowa;300;mysql;185.87.50.40;398890603;380.41 MB“

4.3.4 Relace „DNS“ (#__domains_dns)

Slouží k ukládání informací o konfiguraci DNS záznamů domén. Zdrojem těchto informací je Active 24, s.r.o. a ukládají se zde všechny DNS záznamy mimo SOA DNS záznamy, který mají jinou strukturu, než klasické DNS záznamy.

Tab. 5. Struktura relace „DNS“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
domain_id	Unsigned integer (10)	ID domény
dns_id	Unsigned integer (10)	ID záznamu (Active 24, s.r.o.)
from	Varchar (40)	Začátek platnosti
data	Text	Konfigurace

data_type	Varchar (20)	Konkrétní typ dat záznamu
name	Varchar (100)	Název
to	Varchar (40)	Konec platnosti
ttl	Unsigned integer (10)	TTL
type	Varchar (10)	Typ záznamu

Primárním klíčem je zvolen atribut „id“ a byl zvolen unikátní index na atributu „dns_id“.

Příkladem n-tice relace v relaci „DNS“ může být záznam:

„3;1;14947390;2016-11-30T12:11:53.000Z;dobawebowa.cz.;mailserver;@;2069-12-31T22:59:59.000Z;300;MX“

4.3.5 Relace „SOA DNS“ (#__domains_dns_soa)

Slouží k ukládání informací o konfiguraci SOA DNS záznamů. Zdrojem těchto informací je Active 24, s.r.o. Protože mají jinou strukturu, než klasické DNS záznamy uvedené výše, je vhodné je ukládat do samostatné databázové relace.

Tab. 6. Struktura relace „SOA DNS“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
domain_id	Unsigned integer (10)	ID domény
email	Varchar (100)	E-mail zodpovědné osoby

expire	Integer (10)	Expirace
from	Varchar (40)	Platnost od
dns_id	Unsigned integer (10)	ID záznamu
minimal_ttl	Integer (10)	Minimální TTL
name	Varchar (20)	Název záznamu
name_server	Varchar (100)	Jmenný server
refresh	Integer (10)	Interval kontroly sekundárních DNS oproti primárních DNS
retry	Integer (10)	Doba mezi dotazy sekundárních DNS na primární server při selhání
serial	Varchar (30)	Sériové číslo verze záznamu
to	Varchar (40)	Expirace záznamu
ttl	Integer (10)	TTL

Primárním klíčem je zvolen atribut „id“. Zároveň byl zvolen unikátní index na attributech „dns_id“.

Příkladem n-tice relace v relaci „SOA DNS“ může být záznam:

„14;19;hostmaster.active24.cz.;1209600;2017-02-27T15:27:29.000Z;15558721;3600;@;alfa.ns.active24.cz.;10800;1800;2017022702;2069-12-31T22:59:59.000Z;86400“

4.3.6 Relace „Konfigurace plánů“ (#__domains_plans)

Slouží k ukládání informací o konfiguraci plánů domén (konfigurace kvót). Zdrojem těchto informací je Virtualmin a vypovídá o konfiguracích limitů pro domény. Domény zařazené ve stejných plánech mají vždy nastavené kvóty stejně dle konfigurace plánu.

Tab. 7. Struktura relace „Konfigurace plánů“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
name	Varchar (125)	Název plánu ve Virtualminu
property	Varchar (125)	Název vlastnosti (kvóty)
value	Varchar (125)	Hodnota vlastnosti (kvóty)

Primárním klíčem je zvolen atribut „id“ a opět byl definován unikátní index pro jedinečnost záznamu konfigurace a to na attributech „name“ a „property“.

Příkladem n-tice relace v relaci „Konfigurace plánů“ může být záznam:

„100;Default Plan;maximum_mailbox;Unlimited“

4.3.1 Relace „Rozšíření domény“ (#__domains_extensions)

Slouží k ukládání aktuálně aktivních rozšíření webu, které jsou na webu aplikované. Ukládají se jak CMS, tak i jednotlivá detekovaná rozšíření CMS.

Zdrojem těchto informací je aktualizací logika, která spouští testy na hostingu webu (čtení souborů webu) a pokládá dotazy na databáze webu. Při kladném výsledku testu následně dojde k aktualizaci uložených informací.

Tab. 8. Struktura relace „Rozšíření domény“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
domain_id	Unsigned integer (10)	ID domény
identifier	Varchar (125)	Identifikátor rozšíření
name	Varchar (125)	Název rozšíření
version_svn	Varchar (20)	Detekovaná verze rozšíření instalovaného z SVN

Jako primární klíč je zvolen atribut „id“. Zároveň byl vytvořen unikátní index atributy „domain_id“ a „identifier“ (rozšíření na doméně se může detekovat pouze jednou).

Příkladem n-tice relace v relaci „Rozšíření domény“ může být záznam:

„8469;62;xartcms6;Systeme XartCMS6;3927“

4.3.2 Relace „Historie změn“ (#__domains_histories)

Slouží k ukládání informací o změnách v datech uložených o doméně tak, aby bylo snadné historii ukládat a také ji zobrazovat a číst.

Záznamy neslouží jako body pro obnovu dat v případě reverzních změn, ale k popisu akcí, ke kterým na doméně došlo. Obecně se zde ukládají informace o změnách v seznamu rozšíření domén, konfiguracích domény a konfiguracích DNS záznamů (i SOA DNS záznamu).

Tab. 9. Struktura relace „Historie změn“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
id	Unsigned integer (10)	Jedinečný identifikátor
domain_id	Unsigned integer (10)	ID domény
time	Datetime	Čas vytvoření záznamu
label	Tinytext	Popis akce
state_before	Text	Stav před změnou (JSON objekt)
state_after	Text	Stav po změně (JSON objekt)
type	Set ('added', 'removed', 'info', '')	Systémový typ akce – přidáno, odebráno, informační nebo nedefinovaný

Primárním klíčem je zvolen atribut „id“.

Příkladem n-tice relace v relaci „Historie změn“ může být záznam:

„1018;63;2017-04-28 20:11:44;Extension added;{"identifier":"xartcms6"}“

4.3.3 Relace „Vazby na projekty“ (#__domains_related_projects)

Slouží k ukládání informací o vazbách domén na konkrétní projekty, ve kterých je jistá návaznost. Projekt na doménu může navazovat například tím, že díky projektu tento web vznikl (a i doména) nebo byl projekt zaměřen na implementaci e-shopu, katalogu, atd. Díky implementaci vazeb je pak možné přejít ke konkrétnímu seznamu domén z projektu

(na záložce „domény“) nebo z domény přejít k projektu (přes funkčnost „související projekty“).

Projekt může odkazovat na více domén a více různých domén může odkazovat na stejný projekt (k vazbě více domén na jeden projekt by v praxi nemělo díky politice vedení projektů docházet).

Tab. 10. Struktura relace „Vazby na projekt“, zdroj vlastní zpracování

Název atributu	Datový typ	Popis
domain_id	Unsigned integer (10)	ID domény
related_project_id	Unsigned integer (10)	ID projektu
note	Varchar (255)	Poznámka k vazbě
created_on	Datetime	Datum a čas vytvoření vazby
created_by_id	Unsigned integer (10)	ID uživatele, který vazbu vytvořil

Jako primární klíč je zvoleno spojení atributů „domain_id“ a „related_project_id“ a to díky funkčnosti systému, který modul rozšiřuje (tato struktura je vyžadována).

Příkladem n-tice relace v relaci „Vazby na projekt“ může být záznam:

„114;182;Implementace E-shopu;2017-02-06 08:07:42;592“

4.4 Implementace modulu

Modul se podařilo vytvořit na základě požadavků zadavatelů, předchozích analýz a dostupných prostředků. Poskytuje tedy periodickou automatickou aktualizaci informací o všech doménách na produkčním webovém serveru.

4.4.1 Základní vlastnosti

Mezi informace, které je modul schopen detekovat a uchovávat patří:

- seznam domén (základní informace),
- konfigurace domén,
- databáze domén,
- rozšíření domén,
- DNS záznamy,
- SOA DNS záznamy,
- konfigurace plánů,
- historie změn,
- vazby na projekty.

Tím tedy splňuje základní požadavky společnosti na funkčnost modulu.

Ke kompletní aktualizaci dat dochází 1x/24 hodin v nočních hodinách a to, když se spouští plánovaná systémová úloha.

Modul tvoří celkem 31 761 řádků kódu, vepsaných do 199 souborů, rozmístěných v 36 adresářích.

4.4.2 Oprávnění

Modul ověřuje speciální dvě oprávnění:

- globální přístup k modulu domén,
- globální správa modulu domén.

Globální přístup k modulu domén opravňuje uživatele využívat modul. Bez tohoto oprávnění není modul pro uživatele v systému vůbec dostupný. Toto oprávnění je možné aplikovat pouze uživatelům, kteří mají systémové role „Manažer“ nebo „Člen/Zaměstnanec“. Možnost přístupu k modulu tedy mohou mít uživatelé se systémovou rolí „Administrátor“ (automatický přístup ke všem modulům), „Manažer“ anebo „Člen/Zaměstnanec“.

Globální správa modulu domén opravňuje uživatele měnit popis domény, archivovat informace o doménách a přidružovat domény k projektům. Toto oprávnění je možné aplikovat pouze uživateli, který má systémovou roli „Manažer“. Tato role zároveň automaticky opravňuje přistupovat k modulu (oprávnění „globální přístup k modulu domén“). Možnost správy domén tedy mohou mít uživatelé se systémovou rolí „Administrátor“ nebo „Manažer“.

Uživatelé se systémovými rolemi „Subkontraktor“, „Klient“ a „Klient s omezeným přístupem“ tedy ani nemohou získat oprávnění k přístupu a manipulaci s modulem domén.

Společnost a práva

Extra povolení:

☐ Správa koše — Zaklikněte, pokud chcete, aby tento uživatel viděl položky v koši a mohl je trvale odstranit
 ☐ Správa lidí — Povolit uživateli vytvářet a spravovat existující uživatele a společnosti, krom Administrátorů
 ☐ Správa projektů — Povolit přístup ke všem projektům a povolit uživateli všechna oprávnění k nim
 ☐ Správa požadavků — Povolit uživateli správu požadavků k projektům, které zasílají klienti
 ☐ Použít personální plánování — Umožňuje uživateli použít report Personální plánování, který je jinak omezen pouze pro použití Administrátory.
 ☐ Použít globální dokumenty — Povolit uživatelům přístup ke Globálním dokumentům
 ☐ Správa globálních dokumentů — Povolit uživatelům správu Globálních dokumentů
 ☒ Správa financí — Povolit uživateli správu faktur, plateb, finančních reportů
 ☒ Spravovat cenové nabídky — Toto povolení umožňuje uživatelům spravovat cenové nabídky pomocí modulu Fakturace
 ☒ Správa opakovaných profilů — Toto oprávnění je vyžadováno, pokud chce uživatel v rámci finančního modulu pracovat se smlouvami (Recurring Profiles). Rozšíření od XARTu.
 ☒ Upozornění na opakovaný profil — Toto oprávnění je vyžadováno, pokud chce uživatel používat reporty smluv (Recurring Profiles). Rozšíření od XARTu.
 ☐ Reportovat opakované úkoly — Umožňuje uživateli použít report Opakované úkoly, který je jinak omezen pouze pro použití Administrátory.
 ☒ Evidence mzdových nákladů — Umožňuje uživateli nastavovat hrubou mzdu zaměstnanců a evidovat mzdové náklady na jejich práci.
 ☒ Vyplnit výkaz práce — Umožňuje uživateli použít funkci Vyplnit výkaz práce v modulu Stopky pro rychlé vytváření časových záznamů. Tímto se nemění oprávnění uživatele manipulovat s časovými záznamy.
 ☒ Globální správa modulu domén — Opravňuje uživatele globálně využívat a spravovat modul domén
 ☒ Globální přístup k modulu domén — Opravňuje uživatele globálně využít modul domén

☐ Člen/Zaměstnanec — Člen organizace "XART s.r.o."
 ☐ Subkontraktor — Subkontraktor najatý firmou "XART s.r.o."
 ☐ Klient — Člen firmy klienta s omezenými systémovými právy
 ☐ Klient s omezeným přístupem — Člen společnosti klienta bez přístupu do systému a speciální sadou oprávnění

Uložit změny

Obr. 22. Oprávnění pro uživatelskou roli „Manažer“, zdroj vlastní zpracování

4.4.3 Harmonogram prací pro vytvoření modulu

Modul se podařilo úspěšně zrealizovat, a implementovat do informačního systému podniku a to v termínu od 15. 10. 2016 do 31. 3. 2017.

Realizace modulu probíhala dle následujícího harmonogramu:

Tab. 11. Harmonogram prací pro vytvoření modulu, zdroj vlastní zpracování

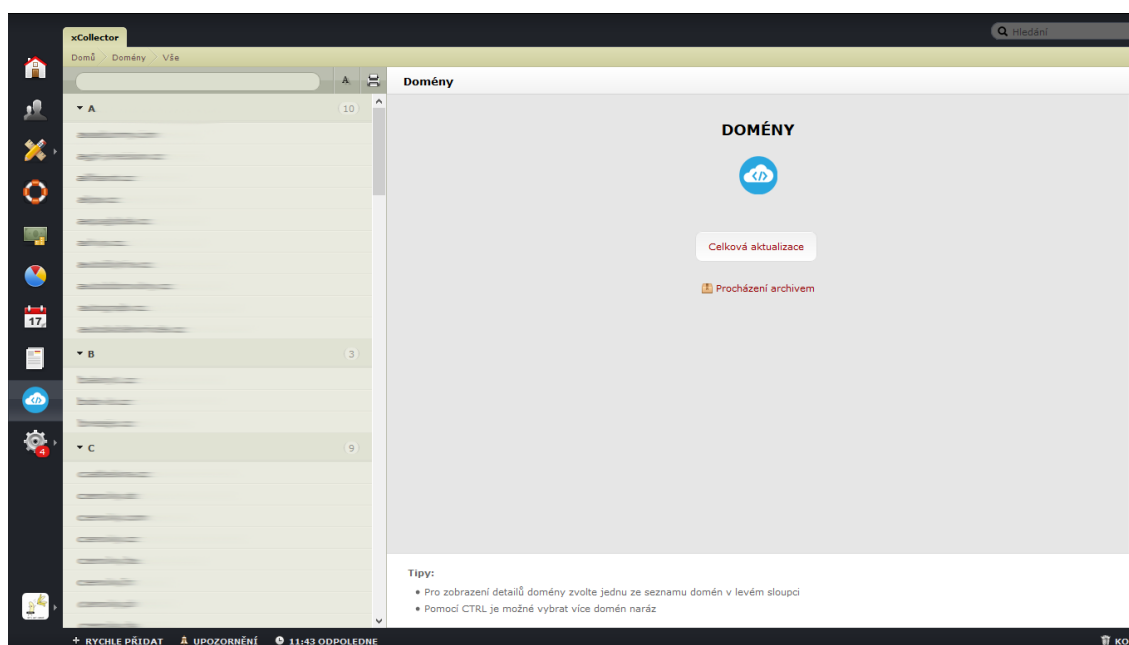
Proces	Začátek procesu	Ukončení procesu
Konzultace zadání, stanovení základních	15. 10. 2016	9. 12. 2016

požadavků na funkčnost modulu		
Příprava technologického zázemí, (klon informačního systému, klon databáze, vytváření přístupů k serveru)	1. 11. 2016	9. 12. 2016
Příprava základu modulu – základní adresářová a souborová struktura	10. 12. 2016	17. 12. 2016
Zpracování procesu ukládání historie změn domény	18. 12. 2016	25. 12. 2016
Zpracování procesu aktualizace seznamu domén	26. 12. 2016	7. 1. 2017
Zpracování procesu aktualizace konfigurace domén	8. 1. 2017	16. 1. 2017
Zpracování procesu aktualizace DNS záznamů domény	17. 1. 2017	28. 1. 2017
Zpracování procesu detekce rozšíření webů (CMS a rozšíření CMS)	29. 1. 2017	12. 2. 2017
Vytváření tříd testů rozšíření webů	13. 2. 2017	27. 2. 2017
Zpracování procesu aktualizace konfigurace plánu domén	28. 2. 2017	9. 3. 2017

Zpracování logiky sdružení domén a projektů	10. 3. 2017	17. 3. 2017
Zpracování pohledů (seznam domén, konfigurace domény)	18. 3. 2017	25. 3. 2017
Přenos modulu do produkční verze informačního systému	26. 3. 2017	31. 3. 2017

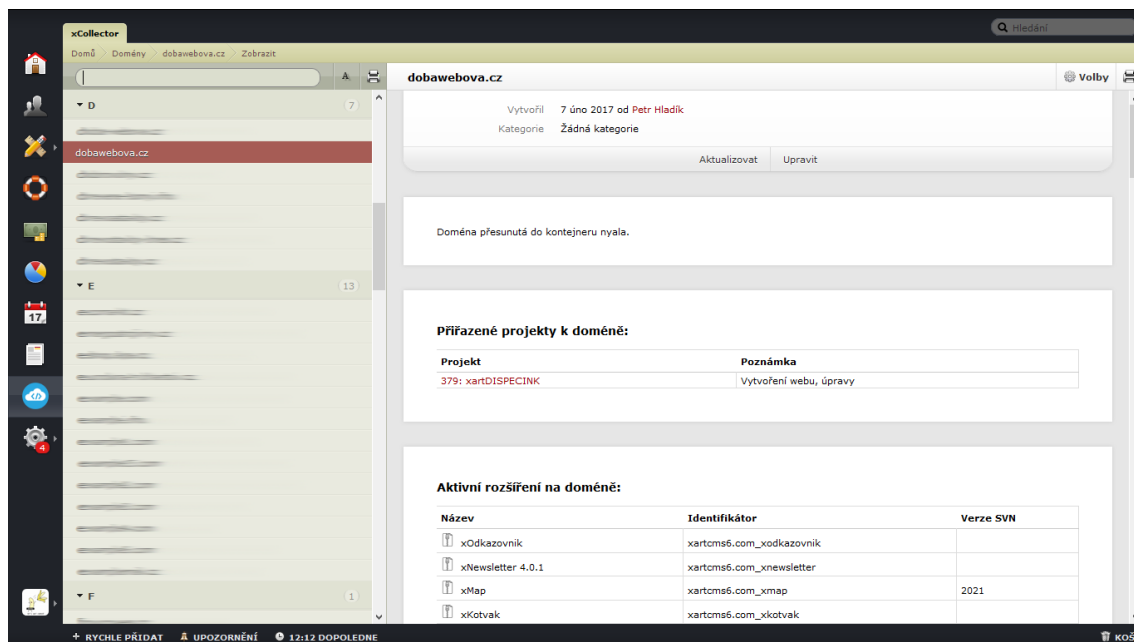
4.5 Vizualizace uživatelského prostředí modulu

Pro vytvoření modulu domén bylo využito standardních systémových prostředků (třídy modulu rozšiřují základní třídy systému), proto i výstup z modulu je ve standardním formátu jako ve zbytku systému (výpis informací modulu domény tedy připomíná do jisté míry výpis informací modulu projektů).



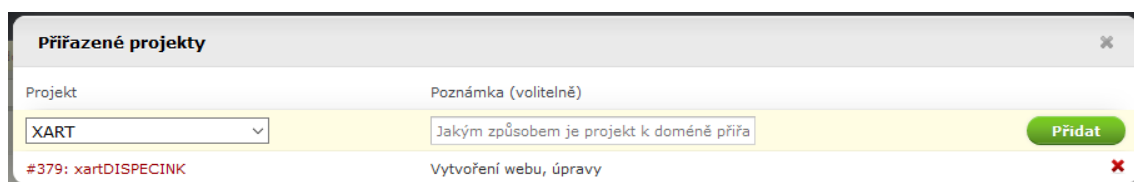
Obr. 23. Vstupní strana modulu – výpis seznamu domén, zdroj vlastní zpracování

Pro výpis přiřazených projektů na detailu domény jsou použity dva sloupce tabulky a to projekt (odkaz do konkrétního projektu) a poznámka (poznámka k vazbě domény na projekt).



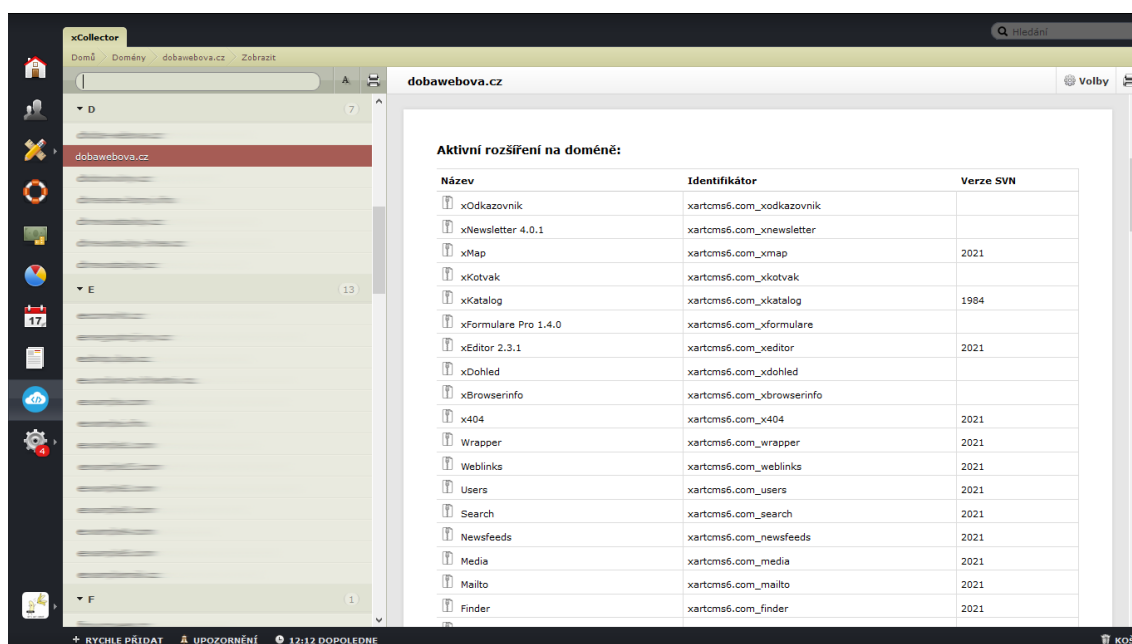
Obr. 24. Detail domény – výpis přiřazených projektů a základních informací domény, zdroj vlastní zpracování

Vazba na projekt lze konfigurovat, pokud uživatel má nastavené oprávnění „Globální správa modulu domén“ v samostatném pohledu.



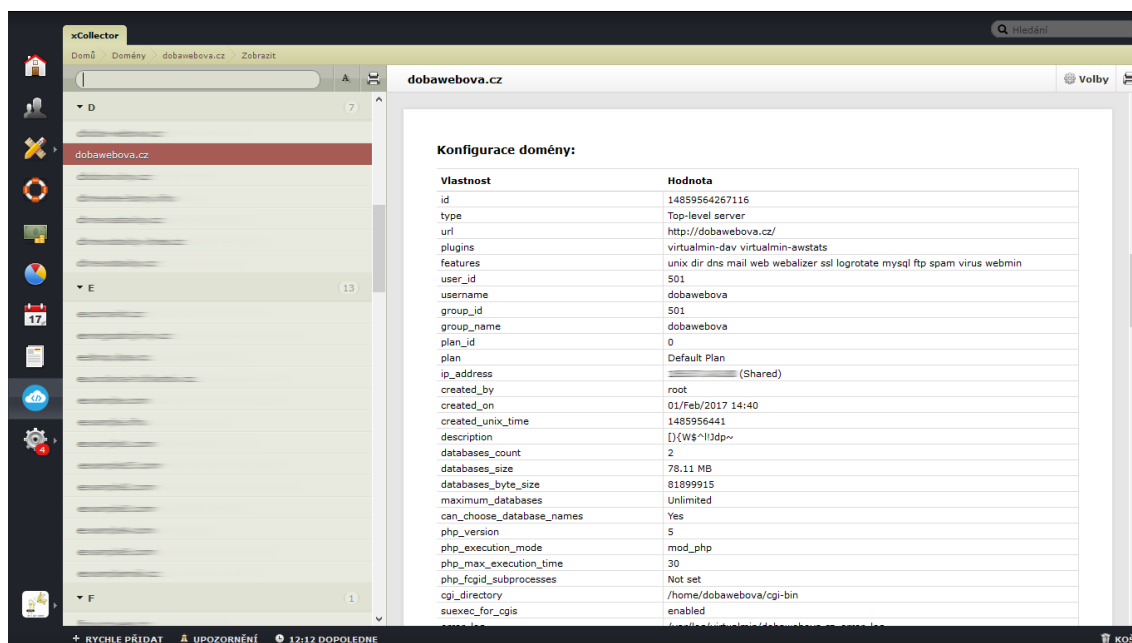
Obr. 25. Přiřazené projekty – konfigurace vazby, zdroj vlastní zpracování

Pro výpis aktivních rozšíření na doméně jsou použity sloupce „Název“ (název rozšíření), „Identifikátor“ (jedinečný identifikátor pro aktualizaci) a „Verze SVN“ (zjištěná nainstalovaná verze při instalaci z SVN).



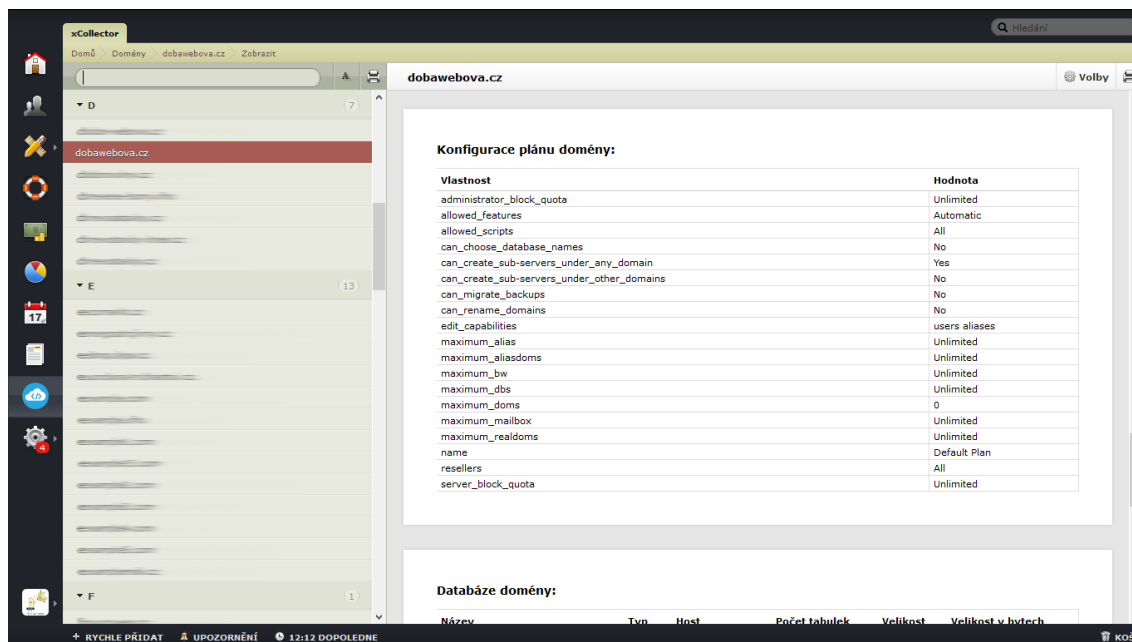
Obr. 26. Detail domény – výpis aktivních rozšíření na doméně, zdroj vlastní zpracování

Pro výpis konfigurace domény jsou použity sloupce „Vlastnost“ (název parametru) a „Hodnota“ (konfigurace parametru).



Obr. 27. Detail domény – výpis konfigurace domény, zdroj vlastní zpracování

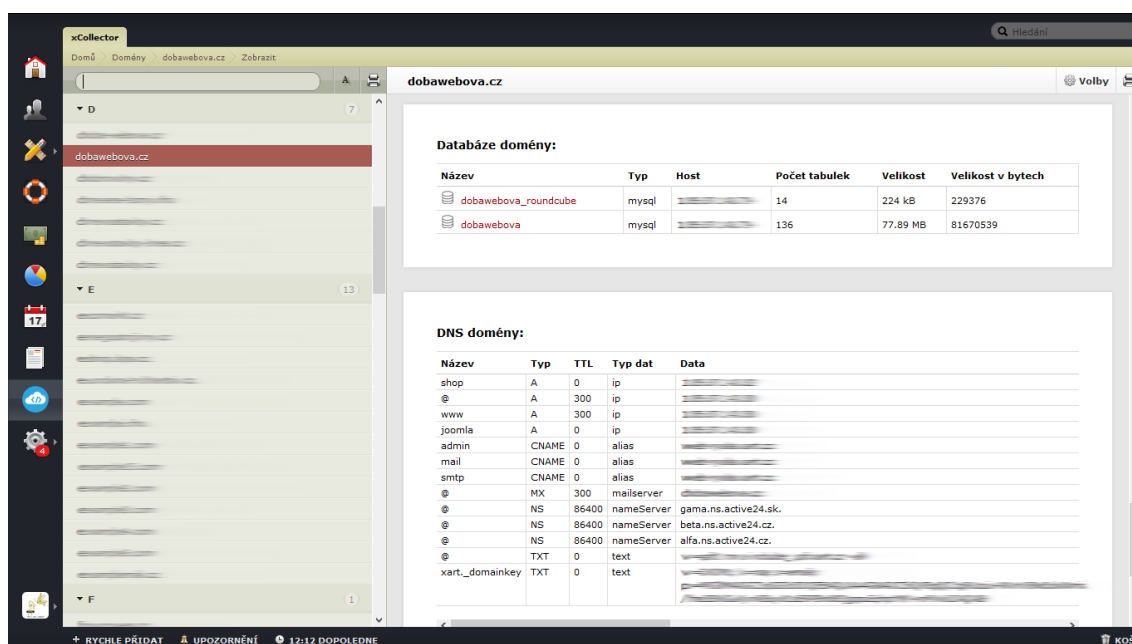
Pro výpis konfigurace plánu domény jsou použity sloupce „Vlastnost“ (název parametru) a „Hodnota“ (konfigurace parametru). Konfigurace konkrétního plánu se u každé domény vypisuje podle toho, jaký plán má doména nastavený ve své vlastní konfiguraci.



Obr. 28. Detail domény – výpis konfigurace plánu přiřazeného doméně, zdroj vlastní zpracování

Pro výpis databází domény jsou použity sloupce „Název“ (název databáze), „Typ“ (typ databáze), „Host“ (IP adresa hostitele), „Počet tabulek“, „Velikost“ (přepočítaná velikost do MB), „Velikost v bytech“ (velikost v B).

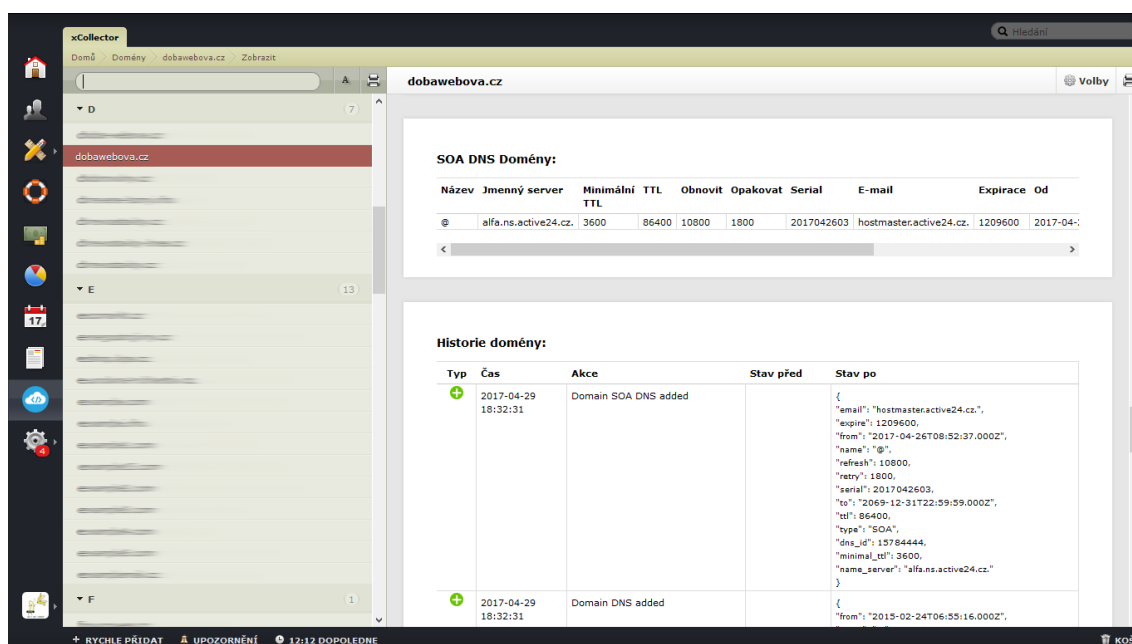
Pro výpis DNS záznamů domény jsou použity sloupce „Název“, „Typ“, „Typ dat“ (název primárního atributu záznamu), „Data“ (hodnota konfigurace).



Obr. 29. Detail domény – výpis databází a DNS záznamů domén, zdroj vlastní zpracování

Pro výpis SOA DNS záznamů domény jsou použity sloupce „Název“, „Jmenný server“, „Minimální TTL“, „TTL“, „Obnovit“, „Opakovat“, „Serial“, „E-mail“, „Exspirace“, „Od“ (kdy přichází záznam v platnost) a „Do“ (kdy končí platnost záznamu).

Pro výpis historie domény jsou použity sloupce „Typ“ (přidáno, odebráno, informace), „Čas“, „Akce“, „Stav před“ (stav před změnou) a „Stav po“ (stav po změně).



Obr. 30. Detail domény – výpis SOA DNS záznamů domén a ukázka historie změn domény, zdroj vlastní zpracování

4.6 Ekonomické zhodnocení

Po dokončení návrhu a implementace modulu v předem dohodnutém rozsahu a termínu, je možné výslednou práci ohodnotit. V této části je uvedeno ekonomické zhodnocení návrhu a implementace modulu domén pro informační systém společnosti.

Jedinou nákladovou položkou v realizaci je odvedená práce, jelikož společnost disponuje všemi technologickými prostředky, které byly pro realizaci potřebné.

4.6.1 Náklady

Realizace práce se řešila v samostatném projektu informačního systému společnosti, kde projekt byl rozdělen na jednotlivé části dle harmonogramu prací a kam byl zaznamenáván i čas strávený na jednotlivých úkolech pro snadnou kontrolu stavu projektu. Díky této skutečnosti můžeme zjistit pracnost jednotlivých částí a následně je i finančně ohodnotit.

Finanční ohodnocení realizací jednotlivých částí lze přepočítat pomocí časové náročnosti a stanovení hodinové mzdy. Hodinovou mzdu pro vyčíslení lze stanovit na 170 Kč.

Tab. 12. Náklady na realizace jednotlivých částí projektu, zdroj vlastní zpracování.

Část realizace	Časová náročnost	Náklady
Příprava základu modulu – základní adresářová a souborová struktura	12,6 h	2142 Kč
Zpracování procesu ukládání historie změn domény	16,35 h	2779,5 Kč
Zpracování procesu aktualizace seznamu domén	23,28 h	3957,6 Kč
Zpracování procesu aktualizace konfigurace domén	18,19 h	3092,3 Kč
Zpracování procesu aktualizace DNS záznamů domény	23,98 h	4076,6 Kč
Zpracování procesu detekce rozšíření webů (CMS a rozšíření CMS)	27,04 h	4596,8 Kč
Vytváření tříd testů rozšíření webů	30,8 h	5236 Kč
Zpracování procesu aktualizace konfigurace plánu domén	18 h	3060 Kč

Zpracování logiky sdružení domén a projektů	15,75 h	2677,5 Kč
Zpracování pohledů (seznam domén, konfigurace domény)	15,26 h	2594,2 Kč
Přenos modulu do produkční verze informačního systému	7,21 h	1225,7 Kč
Celkem	208,46 h	35438,2 Kč

4.6.2 Investice do rozšíření informačního systému

Celkové náklady pro návrh a implementaci nového modulu domén do informačního systému podniku tedy činí 35 438,2 Kč. Tato částka by v dohodnutém rozsahu práce byla konečná a díky automatizaci procesů aktualizace dat nejsou nárokovány žádné administrační náklady.

Podle subjektivních názorů uživatelů informačního systému se díky realizaci modulu domén zkrátila doba pro vyhledávání potřebných informací o klientském webu zhruba o 5 až 15 minut.

Lze tedy říci, že průměrná úspora je nyní 10 minut. Pokud vezmeme v potaz fakt, že denně si zaměstnanec musí dohledávat informace o klientských webech v průměru 4x, po vynásobení s vzorovou hodinovou mzdou zaměstnance, uvedenou v předchozí části, získáváme průměrnou denní úsporu zhruba 113 Kč na jednoho zaměstnance.

4.7 Přínosy pro podnik

Podnik díky implementaci nového modulu do svého informačního systému zautomatizoval doposud zcela manuální proces, kterým bylo zaznamenávání aktuálního stavu klientského webu. Nyní tedy informační systém umožňuje oprávněným uživatelům nalézt aktuální informace v rychlém čase, na jednom místě a interpretované v přehledné formě.

Díky tomu, že jsou tyto informace o klientských webech sjednoceny a uloženy do informačního systému, je možné v budoucnu tento nový modul propojit s ostatními moduly informačního systému. Nabízí se tedy například možnost propojit modul domén s modulem faktur a automaticky vkládat souhrn těchto informací do určitého typu vystavovaných faktur klientům.

Dle kalkulace v předchozí části je úspora vyjádřená v peněžních jednotkách pro vyhledávání informací o klientském webu zhruba 113 Kč. Informační systém podniku denně využívá 11 uživatelů, kteří potřebují získávat informace obsažené v modulu domén. Lze tedy vypočítat, že denní úspora uživatelů informačního systému při vyhledávání informací o klientských webech, vytvořená implementací modulu domén, je celkem 1 243 Kč.

ZÁVĚR

Cílem práce bylo navrhnout a vytvořit nový modul (přehledný nástroj) pro podnikový webový nástroj pro správu projektů, který podnik používá jako podnikový systém. Tento modul má za úkol propojit webové domény klientů, které podnik spravuje, s projekty v projektovém manažeru a to tak, že sám je schopen detekovat a zobrazit skutečný stav hostingu (typ a verze CMS, jeho komponenty a moduly, rozšíření systému, řešení na míru, propojení s API Active 24, s.r.o. a interním SVN).

V této práci byly nejprve definovány nutné obecné předpoklady pro pochopení problematiky, včetně seznámení s podnikovým informačním systémem, a způsobu jakým jsou ukládány ve společnosti informace o webových řešeních. Následně byl konkrétněji popsán očekávaný výsledek práce, na jejímž sestavení se podíleli sami zainteresovaní pracovníci společnosti. Díky tomu mohla být definována očekávaná funkčnost a způsob aktualizace dat, u kterých se požaduje uchovávání v modulu podnikového systému.

V kapitole zabývající se samotným řešením jsou obsaženy postupy při aktualizacích dat, včetně vývojových diagramů pro lepší pochopení a model databázových relací, které modul využívá. Závěrem práce jsou i konkrétní ukázky již hotového modulu s vysvětlením, jak jsou data domény reprezentována uživateli. Prezentován je i časový harmonogram realizace práce s ekonomickým zhodnocením.

V současnosti je modul v plném provozu na produkční verzi podnikového informačního systému a využívají ji hlavně vývojáři společnosti.

SEZNAM POUŽITÝCH ZDROJŮ

- (1) Operační systémy. *Jan na Internet* [online]. 2013 [cit. 2017-05-14]. Dostupné z: <https://www.jaknainternet.cz/page/1757/operacni-systemy/>
- (2) Command Line Interface (CLI). *Techopedia* [online]. 2012 [cit. 2017-05-14]. Dostupné z: <https://www.techopedia.com/definition/3337/command-line-interface-cli>
- (3) Co je Internet a jak funguje? *Datacentrum WEDOS* [online]. Hluboká nad Vltavou, 2010 [cit. 2016-04-01]. Dostupné z: <http://datacentrum.wedos.com/a/17/co-je-internet-jak-funguje.html>
- (4) Historie internetu. *Jak na internet* [online]. 2012 [cit. 2016-04-01]. Dostupné z: <http://www.jaknainternet.cz/page/1205/historie-internetu/>
- (5) Co je IP adresa a jak ji zjistit? *SWMAG.cz* [online]. 2010 [cit. 2017-05-14]. Dostupné z: <http://www.swmag.cz/744/co-je-ip-adresa-a-jak-ji-zjistit/>
- (6) O doménách a DNS. *CZ.NIC* [online]. 2007 [cit. 2017-05-14]. Dostupné z: <https://www.nic.cz/page/312/o-domenach-a-dns/>
- (7) Nastavení DNS záznamů (A, AAAA, MX, CNAME, TXT...). *ACTIVE 24, s.r.o.* [online]. 2017 [cit. 2017-05-14]. Dostupné z: <https://faq.active24.com/cz/866013-Nastavení-DNS-záznamů-A-AAAA-MX-CNAME-TXT>
- (8) Jaké jsou typy DNS záznamů? *GIGASERVER.CZ / Znalostní báze* [online]. 2013 [cit. 2017-05-14]. Dostupné z: <https://kb.gigaserver.cz/jake-jsou-typy-dns-zaznamu/>
- (9) JANOVSKEÝ, Dušan. HTTP Protokol. *Jak psát web* [online]. 2006 [cit. 2016-04-01]. Dostupné z: <http://www.jakpsatweb.cz/server/http-protokol.html>
- (10) KOSEK, Jiří. Co je to HTTP. *VŠE O WWW* [online]. 2008 [cit. 2016-04-01]. Dostupné z: <http://www.kosek.cz/vyuka/4iz228/prednasky/http/foil01.html>
- (11) HEROUT, Tomáš. Co je to internetový prohlížeč. *Help Mark* [online]. 2013 [cit. 2016-04-01]. Dostupné z: <http://www.helpmark.cz/slovníkpojmu/33-internetovy-prohlizec>
- (12) Webová stránka (Internetová stránka). *Management Mania* [online]. 2012 [cit. 2016-04-01]. Dostupné z: <https://managementmania.com/cs/webova-stranka-internetova>
- (13) Definice statické webové stránky. *Myartse.com* [online]. 2014 [cit. 2016-04-01]. Dostupné z: <http://www.myartse.com/definice-staticke-webove-stranky/>

- (14) Hypertext. *IT Slovník* [online]. 2013 [cit. 2016-04-01]. Dostupné z: <http://it-slovník.cz/pojem/hypertext>
- (15) Slovník - HTML. *STARGEN* [online]. Praha, 2009 [cit. 2016-04-01]. Dostupné z: <http://www.stargen.cz/slovník/HTML/>
- (16) ROUSE, Margaret. XML (Extensible Markup Language). *Definition from WhatIs.com* [online]. 2010 [cit. 2016-12-12]. Dostupné z: <http://searchsoa.techtarget.com/definition/XML>
- (17) PHP /základy/. *Tvorba-webu.cz* [online]. 2005 [cit. 2016-04-01]. Dostupné z: <http://www.tvorba-webu.cz/php/>
- (18) VRÁNA, Jakub. Návrh vlastností Zend Engine 2. *PHP triky* [online]. 2009 [cit. 2016-04-01]. Dostupné z: <http://php.vrana.cz/navrh-vlastnosti-zend-engine-2.php><http://php.vrana.cz/>
- (19) VORÁČEK, Jan. *Skriptovací jazyky pro tvorbu webových aplikací*. Pardubice, 2013. Dostupné také z: http://blog.voracek.net/wp-content/uploads/2013/09/JanVoracek_SkriptovaciJazykyProTvorbuWebovychAplikaci.pdf. Diplomová práce. Univerzita Pardubice. Vedoucí práce Ing. Michael Bažant, Ph.D.
- (20) Programovací jazyky. *Programování* [online]. 2015 [cit. 2016-04-01]. Dostupné z: <http://k-prog.wz.cz/progjaz/>
- (21) Programovací paradigma. *Naprogramuj mi.cz* [online]. Brno-sever: MEDIACENTRUM GROUP, 2013 [cit. 2016-04-01]. Dostupné z: <http://naprogramujmi.cz/programovaci-paradigma/>
- (22) Introduction – JavaScript. *MDN* [online]. 2015 [cit. 2016-12-12]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- (23) JavaScript. *Tvorba-webu.cz* [online]. 2006 [cit. 2016-12-11]. Dostupné z: <http://www.tvorba-webu.cz/javascript/>
- (24) Úvod do JSON. *JSON* [online]. 2007 [cit. 2017-05-14]. Dostupné z: <http://www.json.org/json-cz.html>
- (25) Document Object Model (DOM). *W3C* [online]. 1997 [cit. 2016-12-12]. Dostupné z: <https://www.w3.org/DOM/>

- (26) KOSEK, Jiří. PHP a XML. Vyd. 1. Praha: Grada, 2009. ISBN 978-80-247-1116-4.
- (27) ČÁPKA, David. 1. díl - Úvod do objektově orientovaného programování v PHP. *Itnetwork.cz* [online]. 2013 [cit. 2016-04-01]. Dostupné z: <http://www.itnetwork.cz/php/oop/php-tutorial-uvod-do-objektove-orientovaneho-programovani>
- (28) KOFLER, Michael a ÖGGL, Bernd, PHP 5 a MySQL 5: Průvodce webového programátora. Vyd. 1. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9
- (29) KULHAN, Jakub. OOP v PHP. *Programujte.com* [online]. 2009 [cit. 2016-04-01]. Dostupné z: <http://programujte.com/clanek/2009113001-oop-v-php/>
- (30) ČÁPKA, David. 2. díl - První objektová aplikace v PHP. *Itnetwork.cz* [online]. 2013 [cit. 2016-04-01]. Dostupné z: <http://www.itnetwork.cz/php/oop/php-tutorial-prvni-objektova-aplikace>
- (31) VRÁNA, Jakub. 1001 tipů a triků pro PHP. Vyd. 1. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.
- (32) LÁNG, Peter. Co je to framework? *Webárna* [online]. 2007 [cit. 2016-04-01]. Dostupné z: <http://langi.cz/webarna/co-je-to-framework>
- (33) ČÁPKA, David. 1. díl - Úvod do Nette frameworku pro PHP. *Itnetwork.cz* [online]. 2015 [cit. 2016-04-01]. Dostupné z: <http://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>
- (34) ČÁPKA, David. MVC architektura. *Itnetwork.cz* [online]. 2016 [cit. 2016-04-01]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/>
- (35) Databáze. *Misha* [online]. 2011 [cit. 2016-12-12]. Dostupné z: <http://www.databaze.chytrak.cz/>
- (36) HORÁK, M. *Návrh informačního systému*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 75 s. Vedoucí diplomové práce Ing. Petr Dydowicz, Ph.D.
- (37) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství – databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- (38) ŠTUMPF, Jindřich. Webové služby a XML. *SystemOnline.cz* [online]. 2006 [cit. 2017-05-14]. Dostupné z: <https://www.systemonline.cz/sprava-it/webove-sluzby-a-xml.htm>

- (39) What is Webmin? *Webmin* [online]. 1998 [cit. 2017-05-14]. Dostupné z: <http://www.webmin.com/>
- (40) What is Virtualmin? *Webmin* [online]. 2007 [cit. 2017-05-14]. Dostupné z: <http://www.webmin.com/Virtualmin.html>
- (41) Remote API. *Virtualmin* [online]. 2009 [cit. 2017-05-14]. Dostupné z: <https://www.Virtualmin.com/documentation/developer/http>
- (42) Informační systém podniku (Enterprise information system). *ManagementMania.com* [online]. 2012 [cit. 2017-05-14]. Dostupné z: <https://managementmania.com/cs/informacni-system-podniku-enterprise-information-system>
- (43) O ACTIVE 24. *Active24* [online]. 2012 [cit. 2017-05-14]. Dostupné z: <https://www.active24.cz/o-spolecnosti/o-spolecnosti-active24/>
- (44) Zvyšování výkonnosti webu s garancí. *XART* [online]. 2016 [cit. 2017-05-14]. Dostupné z: <https://www.xart.cz/internetovy-marketing/zvysovani-vykonnosti-webu-s-granaci>
- (45) Tvorba webu. *XART* [online]. 2016 [cit. 2017-05-14]. Dostupné z: <https://www.xart.cz/tvorba-webu>
- (46) Datové centrum. *XART* [online]. 2016 [cit. 2017-05-17]. Dostupné z: <https://www.xart.cz/datove-centrum>
- (47) ActiveCollab System Requirements. *Active Collab Classic Help* [online]. 2015 [cit. 2017-05-14]. Dostupné z: <https://activecollab.com/help-classic/books/self-hosted-edition/system-requirements>
- (48) DokuWiki. *DokuWiki* [online]. 2006 [cit. 2017-05-14]. Dostupné z: <https://www.dokuwiki.org/dokuwiki>

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

API – Application Programming Interface – rozhraní pro programování aplikací

B – Byte – násobná jednotka množství informace (bitu)

CGI – Common Gateway Interface – protokol pro komunikaci mezi webovým serverem a programem

CMS – Content Management System – software zajišťující správu dokumentů

CRM – Customer relationship management – řízení vztahů se zákazníky

ERP – Enterprise Resource Planning – plánování podnikových zdrojů

IP – Internet Protocol – základní protokol v internetové síti

MB – Megabyte – násobná jednotka množství informace (bitu)

OOP – Object-oriented programming – Objektově orientované programování

RAM – Random Access Memory – operační paměť

SMB – Server Message Block – síťový komunikační protokol pro sdílený přístup

SMTP – Simple Mail Transfer Protocol – internetový protokol určený pro přenos zpráv elektronické pošty

SSD – Solid-state drive – typ datového média

SVN – Subversion – systém pro správu a verzování zdrojových kódů

TTL – Time to live – číslo, které omezuje dobu platnosti dat

ZFS – Zettabyte File System – kombinovaný souborový systém a správce logických svazků

SEZNAM OBRÁZKŮ

Obr. 1: Životní cyklus dynamicky vygenerované webové stránky s použitím architektury MVC	23
Obr. 2: Základní schéma podnikového software.....	31
Obr. 3: Logo XART, s.r.o.	33
Obr. 4: Ukázka garance splnění měřitelných cílů na ukázkovém anonymním projektu	35
Obr. 5: Tabulka webového řešení	35
Obr. 6: Prostředí informačního systému	39
Obr. 7: Páteří adresářová struktura souborů systému.....	44
Obr. 8: XART Wiki	47
Obr. 9: Zaznamenávání informací pomocí komentářů.....	48
Obr. 10: Zaznamenávání informací pomocí XART Wiki.....	49
Obr. 11: Vývojový diagram procesu aktualizace dat domén	53
Obr. 12: Aktualizace seznamu domén z Virtualminu	54
Obr. 13: Vývojový diagram postupu při aktualizaci seznamu domén	55
Obr. 14: Aktualizace konfigurace domén z Virtualminu	56
Obr. 15: Vývojový diagram postupu při aktualizaci konfigurace domény	57
Obr. 16: Metoda pro aktualizaci aktivních rozšíření domény s dynamickou kontrolou potomků	59
Obr. 17: Vývojový diagram postupu při aktualizaci seznamu aktivních rozšíření webu	60
Obr. 18: Aktualizace DNS záznamů domén od Active 24, s.r.o.	62
Obr. 19: Vývojový diagram postupu při aktualizaci DNS záznamů domény	63
Obr. 20: Aktualizace konfigurací plánů domén z Virtualminu	64
Obr. 21: Relace databázových relací.....	65
Obr. 22: Oprávnění pro uživatelskou roli „Manažer“	77
Obr. 23: Vstupní strana modulu – výpis seznamu domén.....	79
Obr. 24: Detail domény – výpis přiřazených projektů a základních informací domény	80
Obr. 25: Přiřazené projekty – konfigurace vazby	80
Obr. 26: Detail domény – výpis aktivních rozšíření na doméně	81
Obr. 27: Detail domény – výpis konfigurace domény	81
Obr. 28: Detail domény – výpis konfigurace plánu přiřazeného doméně	82
Obr. 29: Detail domény – výpis databází a DNS záznamů domén	83
Obr. 30: Detail domény – výpis SOA DNS záznamů domén a ukázka historie změn domény	84

SEZNAM TABULEK

Tab. 1. Konfigurace serveru pro provoz informačního systému	40
Tab. 2. Struktura relace „Domény“	66
Tab. 3. Struktura relace „Konfigurace domény“	66
Tab. 4. Struktura relace „Databáze domény“	67
Tab. 5. Struktura relace „DNS“	68
Tab. 6. Struktura relace „SOA DNS“	69
Tab. 7. Struktura relace „Konfigurace plánů“	71
Tab. 8. Struktura relace „Rozšíření domény“	72
Tab. 9. Struktura relace „Historie změn“	73
Tab. 10. Struktura relace „Vazby na projekt“	74
Tab. 11. Harmonogram prací pro vytvoření modulu	77
Tab. 12. Náklady na realizace jednotlivých částí projektu.	85

SEZNAM PŘÍLOH

Příloha 1: Výčet navrácených hodnot konfigurací domény z Virtualminu	I
Příloha 2: Adresářová a souborová struktura modulu domén	IV

Příloha 1: Výčet navracených hodnot konfigurací domény z Virtualminu

Navracené hodnoty:

- `allowed_features` – povolené vlastnosti,
- `allowed_scripts` – výčet povolených skriptů,
- `can_choose_database_names` – uživatel smí upravovat názvy databází,
- `can_migrate_backups` – uživatel smí migrovat zálohy,
- `can_rename_servers` – uživatel smí přejmenovat doménu,
- `contact_address` – kontaktní adresa,
- `contact_email` – kontaktní e-mailová adresa,
- `created_by` – název uživatele, který doménu vytvořil,
- `created_on` – datum a čas vytvoření,
- `created_unix_time` – datum a čas vytvoření v unixovém formátu,
- `description` – popis,
- `edit_capabilities` – oprávnění pro úpravy uživatele domény,
- `features` – výčet aktivních vlastností,
- `group_id` – ID skupiny domén,
- `group_name` – název skupiny domén,
- `hashed_password` – přihlašovací heslo v hashovaném formátu (nebudeme v modulu ukládat),
- `home_directory` – cesta ke kořenovému adresáři domény na serveru,
- `id` – identifikátor,

- `ip_address` – IP adresa a informace, zda se jedná sdílenou nebo privátní adresu,
- `login_permissions` – uživatelská oprávnění,
- `mailbox_username_prefix` – prefix uživatele e-mailových schránek,
- `maximum_alias_servers` – maximální počet aliasů,
- `maximum_aliases` – maximální počet aliasů domény,
- `maximum_databases` – maximální počet databází,
- `maximum_mailboxes` – maximální počet e-mailových schránek,
- `maximum_mongrel_instances` – maximální počet instancí Mongerelu (webových serverů),
- `maximum_non-alias_servers` – maximální počet domén, které nejsou aliasy,
- `maximum_sub-servers` – maximální počet subdomén,
- `password_storage` – způsob uchovávání hesla úložiště,
- `plan` – plán, který doméně nastavuje konfigurační kvóty,
- `plan_id` – ID plánu kvót,
- `plugins` – seznam dodatečně povolených pluginů,
- `read-only_mode` – mód pouze pro čtení,
- `shell_command` – nastavení systémového interpretoru spouštění skriptů,
- `shell_type` – typ shellu,
- `sub-servers_cannot_be_under_other_domains` – subdomény nesmí být zařazené pod jinou doménou,
- `sub-servers_inherit_ip_address` – subdomény dědí IP adresu,

- `sub-servers_must_be_under_main_domain` – subdomény musí být zařazené pod hlavní doménou,
- `template` – šablona, podle které byla doména vytvářena,
- `template_id` – id šablony podle které byla doména vytvářena,
- `type` – typ domény (úroveň),
- `user_id` – ID uživatele,
- `username` – přihlašovací jméno.

Příloha 2: Adresářová a souborová struktura modulu domén

